# Calhoun

## Institutional Archive of the Naval Postgraduate School

**Calhoun: The NPS Institutional Archive**

2002-12

# Face recognition using infrared imaging

Pereira, Diogo Camara

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/3471

# NAVAL POSTGRADUATE SCHOOL
## Monterey, California

# THESIS

**FACE RECOGNITION USING INFRARED IMAGING**

by

Diogo Camara Pereira

December  2002

| | |
|---|---|
| Thesis Committee Supervisor: | Monique P. Fargues |
| Thesis Committee Co-Supervisor: | Gamani Karunasiri |
| Thesis Committee Member: | Roberto Cristi |

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | *Form Approved OMB No. 0704-0188* |
|---|---|---|

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE December 2002 | 3. REPORT TYPE AND DATES COVERED Engineer's Thesis |
|---|---|---|

| 4. TITLE AND SUBTITLE: Face Recognition Using Infrared Imaging | 5. FUNDING NUMBERS |
|---|---|
| 6. AUTHOR(S) Diogo Camara Pereira | |

| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000 | 8. PERFORMING ORGANIZATION REPORT NUMBER |
|---|---|

| 9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER |
|---|---|

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| 12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited | 12b. DISTRIBUTION CODE |
|---|---|

**13. ABSTRACT (maximum 200 words)**

This study investigated a (IR) face recognition system using an uncooled IR camera. A computer-based image collection set-up was designed and used to create a small database of 420 facial images, from 14 volunteers. Manual and automated facial image cropping routines were implemented. Two linear approaches for the dataset dimension reduction and classification were implemented and their resulting classification performances compared: PCA-based and LDA approaches. Results show that the best PCA-based average classification performance is equal to 92.22% while the LDA-based classification performance is equal to 99.40%. These results successfully show that an uncooled IR camera may be used to discriminate between individual subjects obtained from a small database collected under a very controlled environment.

| 14. SUBJECT TERMS Face Recognition, Classification, PCA, LDA, IR, Infrared, eigenvectors | 15. NUMBER OF PAGES 115 |
|---|---|
| | 16. PRICE CODE |

| 17. SECURITY CLASSIFICATION OF REPORT Unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified | 20. LIMITATION OF ABSTRACT UL |
|---|---|---|---|

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

i

THIS PAGE INTENTIONALLY LEFT BLANK

**FACE RECOGNITION USING INFRARED IMAGING**

Diogo C. Pereira
First Lieutenant, Brazilian Air Force
B.S.E.E., Instituto Tecnologico de Aeronautica, Brazil, 1994

Submitted in partial fulfillment of the
requirements for the degrees of

**ELECTRICAL ENGINEER**

**and**

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL**
**December 2002**

Author:          Diogo Camara Pereira

Approved by:     Monique P. Fargues
                 Thesis Committee Supervisor

                 Gamani Karunasiri
                 Thesis Committee Co-Supervisor

                 Roberto Cristi
                 Thesis Committee Member

                 John P. Powers
                 Chairman, Department of Electrical and Computer Engineering

iii

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

This study investigated a (IR) face recognition system using an uncooled IR camera. A computer-based image collection set-up was designed and used to create a small database of 420 facial images, from 14 volunteers. Manual and automated facial image cropping routines were implemented. Two linear approaches (PCA and LDA) for the dataset dimension reduction and classification were implemented and their resulting classification performances compared. Results show that the best PCA-based average classification performance is equal to 92.22% while the LDA-based classification performance is equal to 99.40%. These results successfully show that an uncooled IR camera may be used to discriminate between individual subjects obtained from a small database collected under a very controlled environment.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

# LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# ACKNOWLEDGMENTS

I would like to thank my advisors for the guidance and encouragement and I would like to thanks my wife Tatiana for her support and encouragement.

THIS PAGE INTENTIONALLY LEFT BLANK

# EXECUTIVE SUMMARY

This study investigated a (IR) face recognition system using an uncooled IR camera. A computer-based image collection set-up was designed and used to collect a database. The database has 420 facial images, from 14 volunteers (13 adult males and 1 adult female) with 30 pictures per person. All subjects are without glasses. Three different facial expression sets were collected: first, the individuals wore a neutral expression. The second set was constituted of the same subjects with a "smiling" expression. The third set was collected with same individuals pronouncing the vowel "u". Study variables were reduced by using a controlled environment with the same background for each picture, with the person-camera distance fixed, and by restricting the pictures to frontal facial images while allowing a vertical and horizontal angle freedom of $10^{o}$.

The uncooled infrared camera used for imaging has 160 x 120 pixels. Manual and automated facial image cropping routines were implemented with a fixed size of 60 x 45 pixels. Two linear approaches, (a) Principal Component Analysis (PCA) and (b) Linear Discriminant Analysis (LDA) for the dataset dimension reduction and classification were implemented and their resulting classification performances compared. A minimum distance classifier was selected to evaluate the classification performances. The overall system performance was evaluated with a cross-validation scheme using 60% of the pictures for training and 40% of the pictures for testing, with 1,000 repetitions.

Results show that the best PCA-based overall classification performance (92.22%) is obtained when selecting the top 40 eigenvectors, while excluding the first three top eigenvectors. The LDA-based approach performed better, with an overall classification performance equal to 99.40%, as expected from the scheme definition. Results obtained in this study successfully show that an uncooled IR camera may be useful to discriminate between individual subjects obtained from a small database collected under a very controlled environment.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

Infrared (IR) imaging offers advantages over visible imagery, the main one is its invariance to changes due to illumination because the generated images do not depend on reflected light but on radiations produced by the body. The best IR image resolution is produced using cooled camera equipment, where liquid nitrogen is used to reduce the temperature of the sensors. IR face recognition using cooled IR imagery has been used for several years; however, the high associated equipment cost has slowed down its use significantly. However, recent technological developments have resulted in uncooled IR camera set-ups with resolution approaching that of uncooled cameras at a fraction of the cost. This study investigates the design of a (IR) face recognition system using an uncooled IR camera.

First, Chapter II describes the applications of biometrics to everyday life. Chapter III presents infrared imaging concepts and the hardware set-up selected in our study to conduct the experiments. We also describe in this Chapter the image grabbing procedure and the nomenclature selected for the database collection. Chapter IV presents the overall face recognition algorithms implemented for this stud. First, we describe two face-extracting algorithms, which allow isolating the image face-only portions. Next, we present the two dimension reduction schemes considered in this work, Principal Component Analysis (PCA)-based and the Linear Discriminant Analysis (LDA). The minimum distance classifier approach followed to make class decisions. Chapter V presents the recognition performances obtained on our database and discusses the cross-validation approach implemented to estimate the performance. Conclusions and recommendations for future study are presented in Chapter VI. Finally, all code implemented during the study is included in the Appendix.

**THIS PAGE INTENTIONALLY LEFT BLANK**

# II. BIOMETRICS

This section presents the basic human biometrics used today, each with their advantages and drawbacks.

## A. INTRODUCTION

The September 11th terrorist attack clearly demonstrated that security in a broad spectrum of areas needs to be intensified and that the positive identification of individuals is highly desirable in many instances. For example, identification can prevent the unauthorized access to confidential information or physical facilities and can help prevent terrorism and crime.

Biometrics, can certainly increase the probabilities of positive identification by using physical and behavioral characteristics to identify persons or verify identities. Some of the physical and behavioral characteristics used in biometrics are the following:

- Physical Characteristics: chemical composition of body odor; facial features and infrared thermal emissions; features of the retina and iris; fingerprints; hand geometry; wrist and hand veins.

- Behavioral Characteristics: handwritten signature; keystrokes dynamics and voiceprint.

Despite widespread media coverage of biometrics since the September 11th terrorist attacks, a national survey conducted by SEARCH, the National Consortium for Justice Information and Statistics, showed that only half of the general public is aware of such technologies. A public opinion poll was conducted between September 18th and 30th, 2001, shortly after the terrorist attack and again between August 15th and 18th, 2002. According to these surveys, public support for using biometrics for anti-terrorism measures or crime prevention declined from 86% in 2001 to 80% in 2002 [1]. In addition, it is interesting to note that these facts indicate that biometric identification systems remain relatively unknown, yet are generally supported. However, the U.S. Congress has recently passed several laws that are likely to make biometrics far more common. These laws are

only likely to be the inception of an inevitable trend to employ Biometrics for the public's safety. For example:

- The USA Patriot Act, passed in November 2001, requires the Federal Government to develop technology standards for verifying visa applicants identities to ensure that a specific individual has not received a visa under a different name;

- The Enhanced Border Security and Visa Entry Reform Act, which President Bush signed in May 2002, requires that visas and other travel documents include biometric identifiers by October 26, 2004;

- The Aviation and Transportation Security Act, signed in November 2001, authorizes the use of biometrics or other technologies to verify identities of those entering secure airport areas [2].

Furthermore, additional funding is being allocated to develop technologies and policies focused on combating cyberterrorism. For example, the Defense Department awarded Carnegie Mellon University a five-year, $35.5 million grant in that area in September 2002. According to Praddeep Khosla, director of the Center for Computer and Communications Security, the research will focus on four key areas, one of which being computer secured access, and biometrics-based secured access to various devices, and public and private facilities [3].

The International Biometric Association estimates that sales in the biometric industry reached $170 million in 2001, a 70% increase over the previous year. Furthermore, it also predicts that sales will rise to $1 billion in 2004 [4]. According to findings from the international market consulting company, Frost & Sullivan, the total biometric market is projected to reach $2.05 billion in 2006 [5].

Biometrics usually attempt to solve two problems, identification and verification. Identification is the capability to properly identify a person from all those whose biometric measurements have been previously collected and stored in a database. Verification is authenticating a person's identity by comparing the biometrics stored on a database.

Some biometric properties are desirable but are not completely attainable with today's technology. There are biometrics that attain completely some but not all properties. Note that some very accurate biometrics may not be acceptable, which reduces their range of applications. Conversely, some less accurate biometrics may be used as they are easier to implement and are more widely accepted. For example, a gym academy could use a biometric system to restrict its access. It is possible to use a lower performing biometrics that is more acceptable for such an application, where granting incorrect access has limited consequences. However, such a system would not likely be employed in places that require a more controlled access such as nuclear power plants.

## B. BIOMETRICS PROPERTIES

Any human physiological or behavioral characteristic can be used in biometrics, provided it has the following desirable properties:

- Universality: all individuals requiring identification or verification must have their specific biometric characteristic measured;

- Uniqueness: no two persons can possess identical biometric characteristics;

- Permanence: the characteristic must be invariant with time;

- Collectability: the characteristic can be measured quantitatively.

Other requirements are also important, such as:

- Performance: refers to how accurately the biometric data can be identified or verified;

- Acceptability: indicates to what extent people are willing to accept the intrusiveness of the biometric system without feeling their privacy or personal freedoms are being breached;

- Circumvention: refers to how easily the biometrics system can be deceived.

## C. BIOMETRIC TYPES

Biometrics most commonly studied are the face, fingerprint, hand geometry, iris, retinal pattern, signature, voiceprint and thermograms. Note, however, that no single biometric is universal. For example, some individual may have a finger skin too dry to gather fingerprints, as may be the case for a hairdresser routinely working with harsh chemicals [4]. Other individuals may be mute or amputees. The most common biometrics are described as follows:

### 1. Voice

The voice is a unique human characteristic. However, it is usually degraded in quality by microphone, communication channel or analog digital converters used to collect and to transmit information. However, voice acquisition schemes are non-intrusive and many companies, including Bell Canada and Visa International, have tested voiceprint technology. Visa could eventually use this technology to identify credit card holders by having them speak into their PCs microphones to verify online purchases.

Visa has been testing the software Vocent's Voice Secure from Vocent Solutions Inc. of Mountain View, California. Its initial usage has been used solely within the firm, but the company is also considering the idea of deploying the software as a customer-facing application next year [6].

Bell Canada selected Nuance Verifier 3.0 voice authentication software as a security solution for the Speech Enabled Field Access System. This voiceprint authentication technology allows 5,000 Bell Canada technicians to access the dispatch system, and update customer installation and repair orders securely from any telephone [7].

However, voiceprints have their own limitations as reproductions of a recorded voice can circumvent a voice authentication system for a remote unattended application. In such cases, additional security may be obtained by varying the spoken sentence.

## 2. Face

The face is the most acceptable biometrics because it is one of the most common methods of identification, which humans use daily. Furthermore, the method for acquiring samples for verification is non-intrusive. However, developing face recognition techniques that can allow for the effects of aging, facial expressions, slight variations in imaging and photographing variations in the facial pose is quite challenging. A facial-recognition system tested in 2002 at a Palm Beach airport in Florida illustrates the system's limitations. The system failed to match airport employs with their digital photos 53% of the time [8]. Nevertheless, in March 2002 the U.S. Army announced that its military police officers have successfully tested facial-recognition technology to aid them in their duties. The cornerstone of this mobile security system is an eyeglass-mounted wearable camera and display device.

According to Mark Chandler, a physical scientist at the Army's Soldier Systems Center in Natick, Massachusetts, the Army's development of this mobile facial recognition system has been a four-year, $12 million effort. The U.S. Army intends to integrate it with other technologies, and the U.S. Navy is investigating its applications to control access to flight lines [9].

## 3. Fingerprint

Fingerprints are believed to be unique to each person (and each finger). As a result, fingerprinting is one of the most mature biometric technologies used in forensic science worldwide. Fingerprint images are captured, typically, in one of two ways, by scanning an inked impression of a finger or by using a live-scan fingerprint scanner [10].

The FBI inaugurated a fully operational Integrated Automated Fingerprint Identification System (IAFIS) in 1999 with development and implementation costs around $640 million. The IAFIS has its roots in the 1960s and 1970s when the FBI began investigating the feasibility of automating the fingerprint identification process [11,12]. This system compares fingerprints against those contained in its vast database and can respond within two hours. In the past, this process often took weeks to complete using a semi-

automated system.  In addition to criminal checks, the FBI also processes many civil fingerprint submissions. The FBI receives an average of 50,000 fingerprint submissions each business day, where half on those submissions are criminal and half are civil. Note that civil submissions are those for which law requires an FBI fingerprint check, such as for teachers, child care providers, security guards and other such positions of trust. The IAFIS now provides responses to civil submissions within 24 hours, while such requests response times exceeded 100 days before. In addition, fingerprint recognition systems have also been employed for everyday applications, such as for personal computers, grocery stores, etc. For example, customers have the option to pay for their purchases by placing their fingers on a payment terminal at the checkout register at three Kroger stores in College Station, Texas and West Seattle Triftway [13]. The Children's Hospital Inc. in Columbus, Ohio, uses fingerprint technology to authenticate doctors who order tests and prescribe drugs [14].

Semiconductor makers have developed chips that can read a fingerprint and verify a specific user identity. For example, chips made by STMicroelectronics (called the Touch Chip) and Veridicom have already been installed on laptops from MicronPC LLC, Samsung Corp., NEC Corp. and Acer Inc. [15].

### 4. Thermograms

A thermogram measures the temperature of the body's surface, and as a result, the face thermogram may be used as a biometric. Prokoski, a researcher from Mikos Ltd., showed that the temperature pattern on the facial skin is due primarily to the pattern of superficial blood vessels, which are those under the skin but above bone and muscle [16]. Vessels transport warm blood throughout the body and heat the skin. As a result, the skin that is directly above a blood vessel is on the average $0.1^{o}$C warmer than the adjacent skin. The typical human face displays an apparent temperature range of about $8^{o}$C.

The complexity of the resulting 3.1 miles (five kilometers) of bloods vessel in the head and face assures that each person's vascular arrangement is irreproducible and hence unique. Amazingly, studies have shown that even identical twins have different thermogram features [16]. Furthermore, studies have shown that the skin temperature ex-

hibits several periodic fluctuations and that the mechanisms and physiological basis underlying these fluctuations are not yet well understood. Additionally experiments indicate that personality and emotional state are factors that contribute to these fluctuations [17, 18].

## 5. Retinal Scan

The retinal vasculature is rich in structure and is supposed to be a characteristic of each individual and each eye. It is claimed to be the most secure biometrics since it is not easy to alter or replicate. However, many people are suspicious of retinal scans owing to the fact that retinal vasculature can reveal some medical conditions such as hypertension. This fact presently stands in the way of public acceptance of retinal scan based-biometrics.

## 6. Iris

The visual texture of the human iris is determined during the embryonic development and is posited to be unique for each person and each eye [10]. Unlike retinal scans, which require close contacts with the scanner, iris-based recognition has been reported to be successful at distances of up to 18 inches (46 cm) [19]. However, iris scan systems may be disrupted by contact lenses [20].

Iridian Technologies Inc., produces the IrisAcess 2200 system, which detects an individual approaching the imager and then captures an iris image once the person's eye is from three to 10 inches from the mirror on the unit. This image is digitally processed into a 512-byte IrisCode template [21].

The Pentagon tested iris recognition technology at its Athletic Club, as club members may voluntarily sign up to test the system, which involves capturing data from a member's identification card and iris features. In addition, Canada customs will begin using iris scanners to speed air travelers through the country's busiest airports in the near future. Kiosks equipped with iris-recognition devices in eight of Canada's busiest interna-

tional airports will allow some Canadian travelers to cross customs checkpoint in 30 seconds or less by confirming their identities with quick iris scans [22].

## 7. Ear

Studies have shown that the shape of the ear and the structure of the pinna cartilaginous tissue are distinctive. In addition, there is supporting evidence indicating that ears are probably unique to each individual. A. Iannarelli first developed an antropometric ear-based biometric identification technique in 1949. This method is based on 12 measurements of the ear structure. However, this method is not well suited for "machine vision" implementations due to the difficulties of locating the anatomical point that serves as the origin of the measurement system [19]. Although this research is not yet conclusive, studies have also indicated that identical twins have different ear structures (even though they may look similar). [19].

## 8. Gait

Gait refers to the peculiar way one walks and, while it is not claimed to be absolutely unique to each individual, it is sufficiently characteristic to allow for identity authentication. Naturally, gait is a behavioral biometric and may not stay invariant especially over a long period of time, due to large fluctuations of body weight, major injuries involving the joints or cerebral disorders or inebriety. Typically, gait features are derived from analyzing video-sequence footage of a person walking and encompass several characteristic movements of each articulate joint.

Gait recognition research has largely been motivated by Johansson's experiment and the ability of humans to perceive motion from Moving Light Displays (MLDs). In these experiments, human subjects were able to recognize the type of movement of a person by observing the 2D motion pattern generated by light bulbs attached to the person. Similar experiments, later showed evidence that the identity of a familiar person, as well as the gender of the person might be recognized from MLDs [23].

### 9. Keystroke Dynamics

This behavioral biometrics is also not expected to be unique to each individual, but it may offer sufficient discriminatory information to permit for identity authentication. Keystroke dynamic features are based on time durations between the keystrokes. Some variants of identity authentication use features based on inter-key delays as well as dwell times—namely, how long a person holds down a key.

### 10. DNA

DNA (Deoxyribonucleic Acid) is mostly used in forensics for identification and is believed to be unique for each person (except for identical twins who have identical DNA patterns). Three issues limit the utility of this biometric for other applications [10]:

- Contamination and Sensitivity: it is easy to steal a DNA sample for a fraudulent or criminal purpose;

- Automatic Real-time Identification Issues: presently, the technology for genetic matching for online identification is lacking. Most of the human DNA is identical for the entire human species and only some relatively small numbers of specific locations on DNA, called polymorphic loci, exhibit individual variations. These variations are manifested either in the number of repetitions of a block of base sequences or in the minor non-functional perturbations of the base sequence. The processes involved in DNA-based personal identification determine if two DNA samples originate from the same individual based on the distinctive signature at one or more polymorphic loci. The major components of these processes now exist in the form of chemical methods requiring an expert's skills.

- Privacy Issues: information about susceptibilities of a person to certain diseases could be gained from the DNA pattern, and there is a significant concern that the unintended abuse of genetic code information may result in discrimination, such as in hiring practices.

## 11. Signature

The way a person signs his or her name is known to be a characteristic of that individual. However, signatures are not a foolproof behavioral biometric as they evolve over a period of time and are influenced by physical and emotional conditions. Although experts can discriminate between genuine and forged signatures, modeling the invariance in the signatures and automatic signature recognition process are challenging. Two approaches to signature verification exist, static and dynamic. In the static signature verification, only geometric features of the signature are used for authentication. In dynamic signature verification, the signature dynamic features such as acceleration, velocity, and trajectory profiles are also employed.

A related, and remarkable, technology is authentication based on the characteristics of the sound produced during a signature. These acoustic emissions are claimed to be a characteristic of each individual [10].

## 12. Odor

It is commonly known that body odors serve several functions including communication, attracting mates, assertion of territorial right, and protection from a predator. Humans produce an identifiable odor that is characteristic of their unique chemical composition. A component of the odor emitted by human (or any animal) is distinctive to a particular individual. Despite the rare use of this technology in biometrics, there are many commercial electronic "noses" and software that can recognize odors under development. Generally, these common applications are in food and beverage processing, environmental monitoring, medical diagnostic, and fragrance development [24, 25].

The artificial nose manufactured by SMart Nose Inc., uses mass spectrometry to identify volatile organic components ranging from liquid to solid samples. The instrument is fully automated and operates in two modes: a survey mode for the setup of new applications, and a multiple channel mode for the specific and/or sensitive analyses in the day-to-day use. The system uses the algorithm based on the Principal Component Analysis (PCA) or Discriminant Function Analysis (DFA) [24].

### 13. Hand and Finger Geometry

Some features related to a human hand such as finger length are relatively invariant and peculiar (although not unique) to each individual [10]. Hand geometry readers have been the dominant biometric technology for access control and time and attendance applications in 2001, as reported by Frost & Sullivan's World Biometric Report 2002 [26]. Since 1991, HandReaders from IR Recognition Systems, have produced more than 100 million biometric verifications, with more than 50,000 produced on high volume days at the San Francisco International Airport. HandReaders span the entire airport, securing more that 180 doors and verifying the identity of more than 18,000 employees.

Over 70,000 HandReaders are installed throughout the world in a wide variety of applications. The 1996 Olympic games used HandReaders to protect access to the Olympic Village, where more than 65,000 people were enrolled and over 1 million transactions were handled in 28 days [26].

### D. COMPARISON AMONG BIOMETRICS

Biometrics may be compared in relation to their desirable properties. Table 1 provides a comparison of biometrics technologies.

Table 1.    Biometric Technologies Comparison. The data was obtained from [10].

| Biometrics | Universality | Uniqueness | Permanence | Collectability | Performance | Acceptability | Circumvention |
|---|---|---|---|---|---|---|---|
| Face | High | Low | Medium | High | Low | High | Low |
| Fingerprint | Medium | High | High | Medium | High | Medium | High |
| Hand Geometry | Medium | Medium | Medium | High | Medium | Medium | Medium |
| Keystrokes | Low | Low | Low | Medium | Low | Medium | Medium |
| Hand Vein | Medium | Medium | Medium | Medium | Medium | Medium | High |
| Iris | High | High | High | Medium | High | Low | High |
| Retinal Scan | High | High | Medium | Low | High | Low | High |
| Signature | Low | Low | Low | High | Low | High | Low |
| Voice Print | Medium | Low | Low | Medium | Low | High | Low |
| Face Thermograms | High | High | Low | High | Medium | High | High |
| Odor | High | High | High | Low | Low | Medium | Low |
| DNA | High | High | High | Low | High | Low | Low |
| Gait | Medium | Low | Low | High | Low | High | Medium |
| Ear | Medium | Medium | High | Medium | Medium | High | Medium |

This Chapter provided an overview of the use of biometrics today with their own advantages and drawbacks. Next, we discuss the main concepts behind infrared imaging.

# III. INFRARED IMAGING

This chapter presents the infrared imaging system used. First, we address relationships between the temperature of a source and the infrared radiation. Second, we present the camera and physical set-up used to obtain the images.

## A. INFRARED SPECTRUM

The astronomer Sir William Herschel (1738 – 1822) was the first to detect infrared (IR) radiation in 1800. IR radiation covers 0.7 to 1000 μm in the electromagnetic spectrum. According with Hecht, the IR spectrum is subdivided into four regions: the near IR (0.78 – 3 μm), the intermediate IR (3 – 6 μm), the far IR (6 – 15 μm) and the extreme IR (15 – 1000 μm) [27].

## B. BLACKBODY

A blackbody is a perfect radiator. It radiates more power from a surface area in a wavelength interval than any body can radiate at a given temperature. No surface that is in thermodynamic equilibrium can radiate more power, unless it contains fluorescent or radioactive materials.

The expression for the blackbody emission was first formally derived by Planck in 1900. The exitance for a blackbody, $M(\lambda,T)$, considering the irradiation in a hemisphere in front of the source, is given by [28]:

$$M\left(\lambda,T\right)\left[\frac{\text{watt}}{\text{cm}^2 \cdot \mu\text{m}}\right] = \frac{2\pi hc^2}{\lambda^5[e^{hc/\lambda kT}-1]},\qquad(3.1)$$

15

where $h = 6.625 \times 10^{-34}$ Watts $\cdot$ s$^2$ is the Plank's constant, $k = 1.38 \times 10^{-23}$ Watts$\cdot$s$\cdot$K$^{-1}$ is the Boltzman's constant, $c = 3 \times 10^8$ m/s is the speed of light, and $T$ is the temperature in Kelvin. $M(\lambda,T)$ is usually expressed in Watt $\cdot$ cm$^{-2}$ $\cdot$ µm. Figure 1 plots the exitance as a function of the wavelength, for three different temperatures near room temperature (300K).
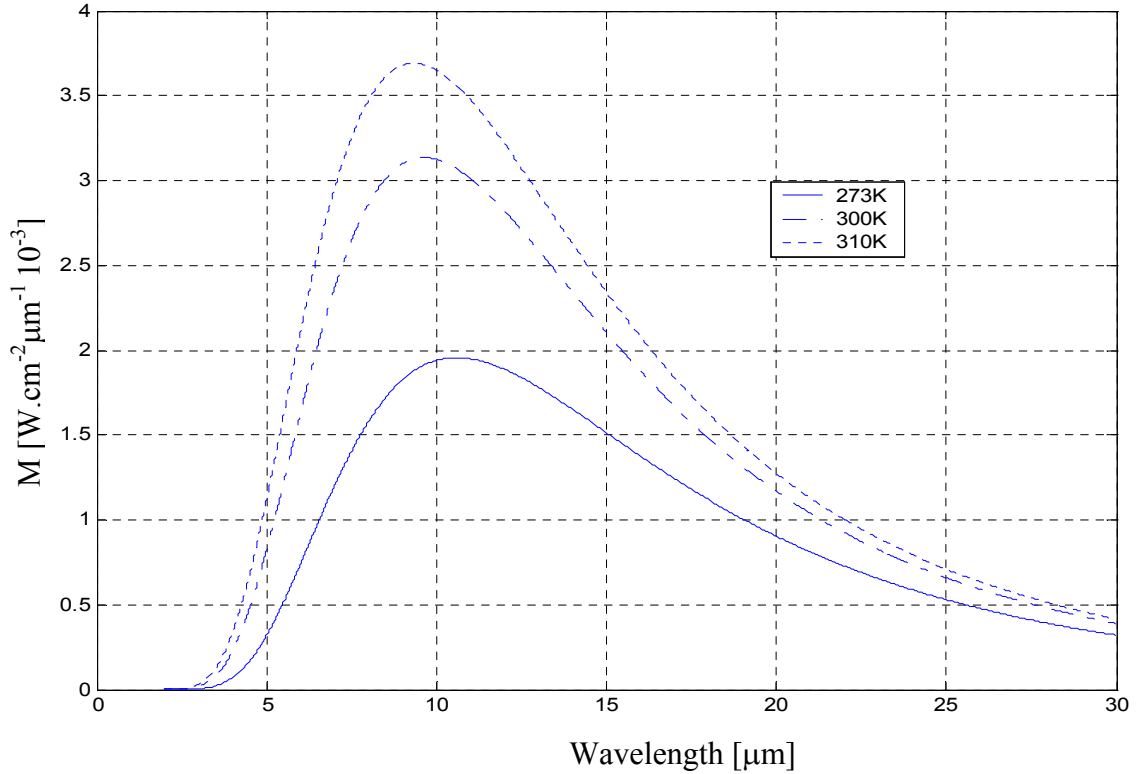


Figure 1.    Blackbody exitance.

The Wien displacement law defines the wavelength that corresponds to the peak of the exitance at each temperature. This can be obtained by computing the partial derivative, with respect to wavelength, of the exitance and setting it equal to zero, expressed as:

$$\frac{\partial M(\lambda,T)}{\partial \lambda} = 0, \qquad (3.2)$$

which leads to $\lambda_{max} T = 2898 \, \text{µm} \cdot \text{K}$, where $\lambda_{max}$ is the wavelength where the exitance peak occurs at the temperature $T$. The maximum exitance occurs at $\lambda = 9.66$ µm for a blackbody source at $T = 300$ K (room temperature), while the maximum exitance occurs at $\lambda = 9.35$ µm for a blackbody source at $T = 310$ K (human body temperature).

16

The total blackbody exitance is obtained by integrating the exitance expression over a finite spectral region, which leads to:

$$M = \int_{\lambda_1}^{\lambda_2} M(\lambda,T) \, d\lambda, \qquad (3.3)$$

where $\lambda_1$ and $\lambda_2$ are the smallest and largest wavelengths of the finite spectral region of interest.

The total exitance from a blackbody at temperature $T$ is the integral over all wavelengths values, and is given by:

$$M(T) = \int_{0}^{\infty} M(\lambda,T) \, d\lambda, \qquad (3.4)$$

which leads to $M(T) = \sigma_e T^4$, where $\sigma_e$ is the Stefan-Boltzmann constant which has an approximate value of $5.67 \times 10^{-12}$ watt/(cm$^2 \cdot$ K$^4$).

## C. EMISSIVITY

The previous section presented the equation of the blackbody exitance. Recall that the blackbody is a perfect radiator as it radiates the maximum power from a surface area in a wavelength interval. So, it also represents an upper bound for the overall exitance of a source.

The ratio between the exitance of a real source and the exitance of a blackbody is defined as the emissivity $\varepsilon$, which is a function of the wavelength $\lambda$ and the absolute temperature T. It is a dimensionless quantity smaller than 1 and defined as:

$$\varepsilon(\lambda,T) = \frac{M(\lambda,T)_{source}}{M(\lambda,T)_{blackbody}}. \qquad (3.5)$$

Note that a blackbody source has an emissivity $\varepsilon$ equal to 1 for all wavelengths, while a graybody source has an emissivity less than 1 for all wavelengths.

17

Human skin emissions can be approximated by a graybody source with ε equal to 0.98 in the wavelength range between 3 and 14 μm [29]. For a typical person, temperature across the face varies about 8°C [16]. Since this variation is much smaller than the average absolute temperature ($T_{av}$) of the face (310 K), the variation of the exitance across the face can be taken expressed as:

$$\Delta M \simeq 4\sigma_e T_{av}^3 \Delta T .$$ (3.6)

For example, the face exitance changes by approximately 1% when the face temperature changes by 1%.

## D. INFRARED IMAGES

### 1. IR Camera

The uncooled IR camera selected for this study is a IR-160 model produced by Infrared Solutions, Inc., where the primary reason for using an uncooled camera was to reduce the overall system cost. This uncooled camera is sensitive to electromagnetic radiation in the wavelength range from 8 μm to 14 μm. It has a 160 x 120 pixel microbolometer array to obtain the images at a 30 Hz frame rate that can be displayed on a NTSC standard monitor, or may be transmitted serially using 8 bits per pixel via RS-232. Microbolometers operate on the principle that temperature changes produced by the absorbed radiation produce changes in the material resistance [28]. As a final result, an electrical signal proportional to the change in the electrical resistance is obtained. The bolometer temperature coefficient depends on the sensor material and the amount of resistance change produced by a temperature change. The relation between the temperature and the resistance may be approximated by the linear equation:

$$R = R_0 (1 + \alpha \Delta T_d) ,$$ (3.7)

where $\alpha$ is the temperature coefficient of the detector material, $R_o$ is the resistance at the base temperature and $\Delta T_d$ is the change in the detector temperature. The camera has a germanium lens that allows for manual focusing of the image. In addition, it is possible to adjust the image brightness and contrast and to select a specific palette scheme. Selec-

tions are saved in non-volatile memory and the camera initiates with the same setting the next time it is turned on. The palette scheme used in our study is linear, which allows us to adjust the gray color displayed automatically. In addition, it sets the output image pixel that corresponds to the image pixel with the highest and lowest intensity to white and black, respectively. The camera is remotely controlled using the program HyperTerminal from Windows and its focus is manually adjusted for each picture until the best setting is achieved. In our work, all images were digitized using eight bits per pixel and contained one face only. The noise equivalent difference temperature (NEDT) of the camera is about 60 mK. The transmission time per frame using the RS-232 interface is about 6 seconds.

## 2. Image Acquisition

The image acquisition scheme is illustrated in Figures 2 and 3. There were nine numbered points marked on the walls. Numbers vary from 1 to 9, as depicted in Figure 3. Each volunteer was asked to rotate his or her head in the direction of the numbers. An additional picture was taken by asking volunteers to look at a random place within the square formed by the extreme marked points. All volunteers were adults (13 males and 1 female), all males were clean-shaven except for 1 subject with a mustache. For each volunteer three sets of images with different expressions were obtained. Each set contained 10 images, resulting in 420 images. A sample of the images obtained is shown in Figure 4.
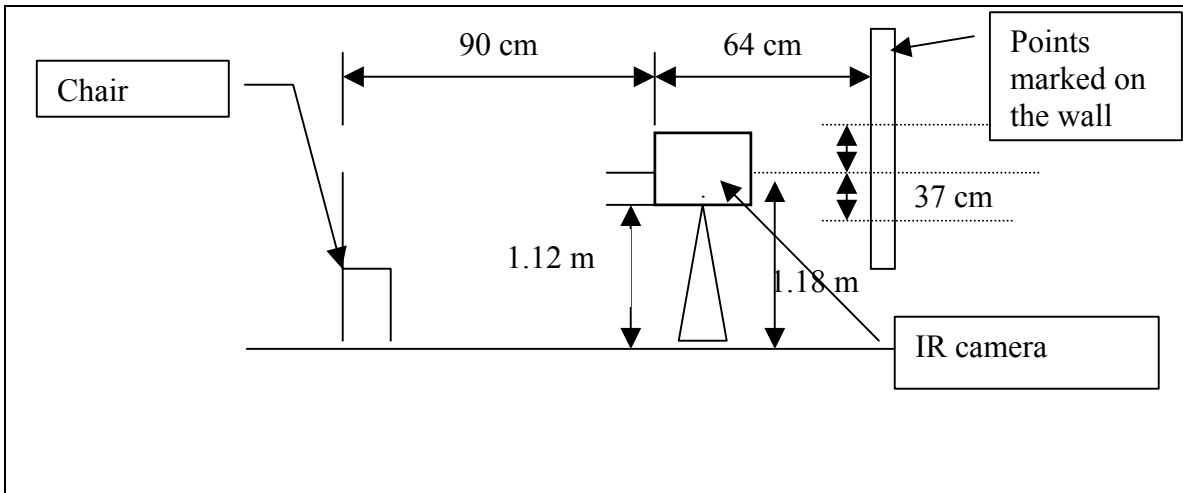
Figure 2.    Image acquisition (lateral view).
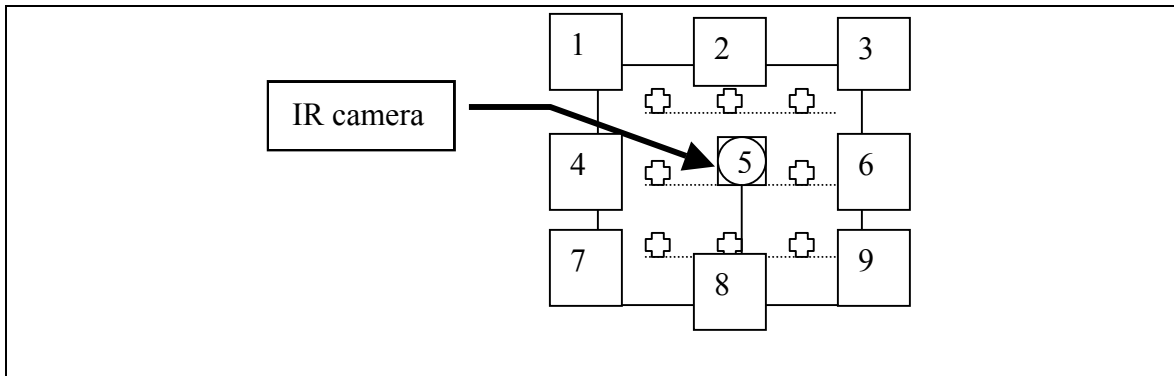


Figure 3.    Image acquisition (front view).
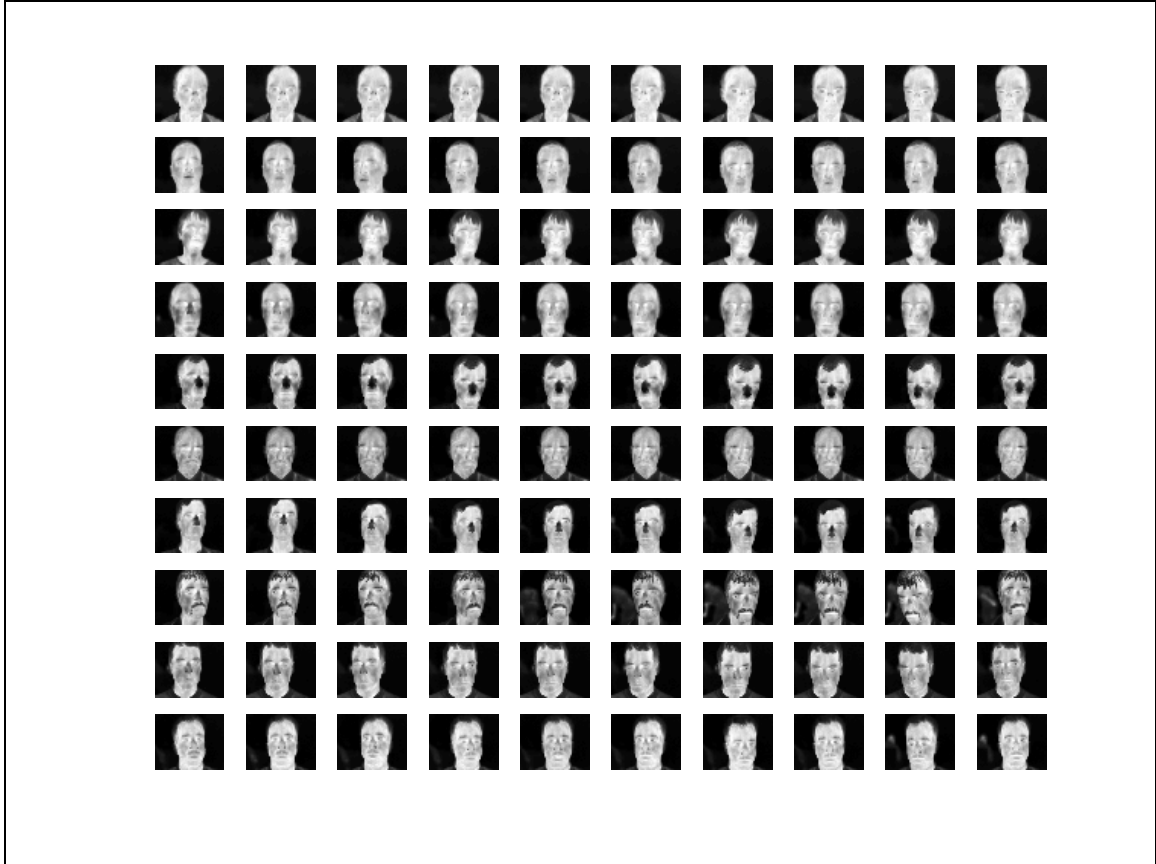
Figure 4.    Sample of the images obtained.

## 3. Image Nomenclature Convention

The images obtained were stored in .pgm format using the following nomenclature: *xx-yy-zz.pgm*, where *xx* is the person number, *yy* represents the visual direction that the person should be looking at and *zz* is the section number.

The range of parameters selected for this study is:

- [1–6,8,9,11–16] for *xx*,

- [1–10] for *yy*,

- [1,5,6] for *zz*.

The section number (zz) was defined as follows:

1- for a neutral expression and no glasses.

21

5- for  a "smiling" expression and no glasses.

6- for a vowel "u" pronouncing expression and no glasses.

For example, *2-5-5.pmg* refers to subject number 2, looking at visual direction 5, with a "smiling" expression and no glasses.

This Chapter presented the infrared imaging concepts, the setup used to collect the images and the image nomenclature convention. Next chapter discusses the classification schemes investigated.

# IV. FACE RECOGNITION

Automatic face recognition is a procedure for associating a face to an identity. We may say that a face is recognized when its image matches reasonably well the face image of someone whose identity we know. This procedure may be divided in three sequential steps: a) Image acquisition; b) Face detection and segregation; and c) Face classification, as shown in Figure 5.

| Image from IR camera | → | Face detection and segregation | → | Classification |
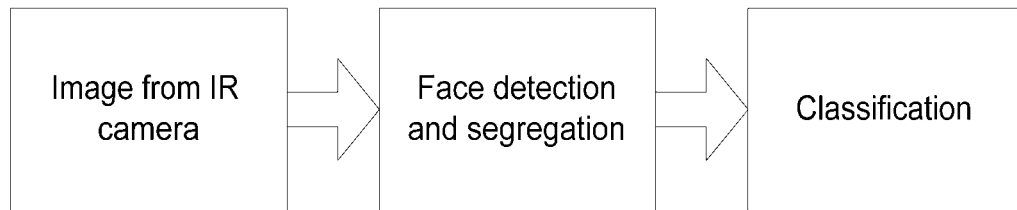
Figure 5.    Automatic face recognition procedure block diagram.

## A. FACE DETECTION AND IMAGE SEGMENTATION

IR images were collected with the IR-160 camera, as described in Chapter II. First, the images were cropped to isolate face-only portions to minimize the potential impact due to different backgrounds. Next, the resulting cropped images were used in the training and testing phases of the automatic recognition system. Two methods for cropping were used, manual cropping and automatic cropping. These methods are described next.

### 1. Manual Face Detection and Segmentation

In this work, the face contained in each picture was detected by visual inspection and manually cropped using a rectangle of 60 vertical by 45 horizontal pixels. The size of this rectangle was obtained by trial and error so that it included most of the facial features of each subject, leaving very little visible background. Faces were cropped just above the chin and above the eyebrows. Next, each cropped image was saved in *.bmp* format using the same nomenclature as the image it was obtained from, except that an "-a" was ap-

pended to the file name, such as *2-3-4-a.pgm* for example. A sample cropped image is shown in Figure 6 below.



Figure 6.    Cropped IR image.

## 2. Automatic Face Detection and Segmentation

An automated face detection and cropping routine was also implemented to decrease the manual processing time.  It operated in two steps:

- First, images are vertically cropped using a basic edge detection scheme.

- Second, a fine search on the resulting image is conducted to extract the face-only portion of the image.

### a. Face Edge Location

Face edge location is conducted in the following three steps. First, a "difference" image is generated by subtracting a shifted copy of the image from the original image, resulting in an "edge enhanced" image, as shown in Figures 7 and 8 below. Pixel values obtained for row 60 of the difference image are plotted in Figure 9, which shows that vertical left and right edges may be easily located by extracting the indices corresponding to the maximum (~40) and minimum (~117) values respectively. The overall vertical left and right face edges are established by selecting the smallest left edge index and the largest right edge index out of three rows selected in the image. The resulting cropped image is the original image cropped on the right and left edges of the face, as illustrated in Figure 10.
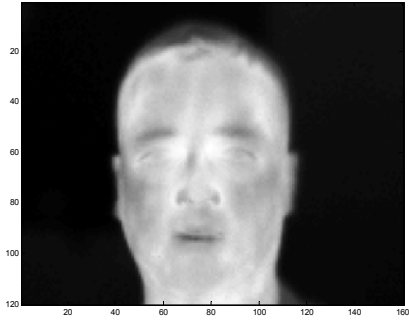
Figure 7.    Infrared image.
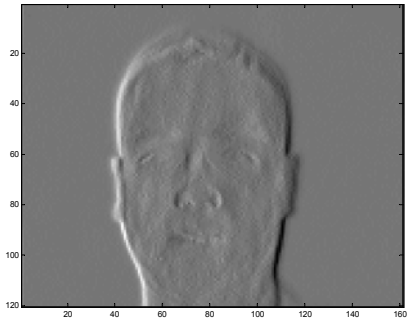


Figure 8.    Difference image obtained from Figure 7.
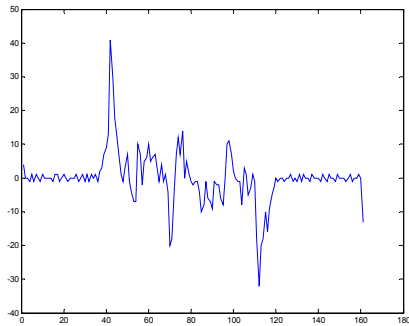


Figure 9.    Row number 60 pixel values obtained from Figure 8.
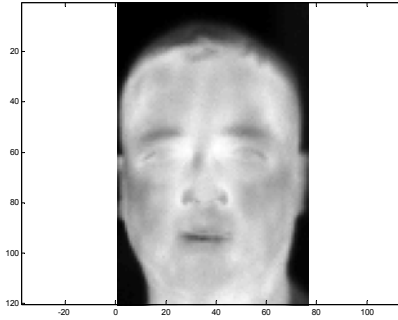
Figure 10.    Automatically cropped image.

### b. Face Locating Procedure

At this stage we still need to further crop the top and bottom portions of the cropped image to isolate the face-only section and remove any left-over background. Turk and Pentland proposed a procedure to extract exact face location contained in an image [30]. Their algorithm uses the Principal Component Analysis (PCA) approach, which is explained in further details in the next section, and considers the Euclidian distance as a measure of the "closeness" of two images. Basically, their procedure divides the image into sub-images, projects them using PCA into a predefined "face space" and uses the fact that "face-only" sub-images are closer to the face space than non "face-only" subimages are. The procedure is described next.

First, we extract sub-images $\Gamma$ from an image $I$ and subtract the mean image of the whole database $\Psi$, as the mean does not contain any useful information for this task. The resulting mean centered sub-images are called $\phi$. Next, the mean-centered sub-images are projected onto the face space, which leads to their associated sub-images $\phi_f$. At that point, we use the Euclidian distance to evaluate how close each sub-image is from its projection onto the face space and select as the "face-only" sub-image that which leads to the minimum distance $\varepsilon^2$ out of all sub-images extracted from that image. Note that this method is computationally expensive.  However, we can obtain the face map directly from the original image without explicitly applying the projection operation, as shown in [30, Eq. 15]. The distance expression $\varepsilon^2$ can be shown to be equal to:

26

$$\varepsilon^2(x,y) = \Gamma^t(x,y)\Gamma(x,y) - 2\Gamma(x,y) \otimes \Psi + \Psi^t\Psi - \sum_{i=1}^{L}[\Gamma^t(x,y) \otimes u_i - \Psi \otimes u_i]^2, \qquad (4.1)$$

where $\Gamma$ is the subimage, $\otimes$ is the correlation operation, $\Psi$ is the mean image of the whole database used for the system design and $u_i$ are the eigenvectors of the covariance matrix used to create the face space, as defined in the PCA section later. The computation of the face map involves only the computation of $L+1$ correlation terms over the input image, and the computation of $\Gamma^t(x,y)\Gamma(x,y)$. Computing the expression $\Gamma^t(x,y)\Gamma(x,y)$ is obtained by squaring the input image $I(x,y)$ and summing the squared values contained in each subimage. This procedure was implemented in MATLAB and the code is included in the Appendix.

## B. FACE CLASSIFICATION

Classification aims to assign an identity to an input facial image by comparing it to pre-stored facial images. A direct image-by-image comparison would be extremely computationally expensive. Therefore, a dimension reduction process, which reduces the search time and memory required to store all subjects, is very desirable. Note that an image may be considered as a vector of dimension equal to the number of pixels in the image by reshaping the image as a vector. For example, consider a 60 by 45 pixels image, which contains a total of 2,700 pixels. As a result, the image vector produced by this image is of length 2,700. In addition, pixels are likely to be correlated. As a result, a dimension reduction scheme is likely to reduce computational load and storage requirements without degrading discrimination performances. In this work, we consider two linear projection approaches which allow us to significantly reduce the problem dimension: the Eigenface approach, based on the Principal Component Analysis (PCA) concept, and the Fisherface approach, based on the PCA and Fisher Linear Discriminant (FLD) concepts.

27

## 1. Principal Component Analysis (PCA)

### a. Definition

PCA is a linear projection scheme that reduces the dimensionality of a data set while retaining as much variance present in the data as possible. As a result, the PCA scheme is best matched for compression applications, but has also been applied with success to classification problems, although one cannot insure the components kept are those best suited for discrimination purposes [31].

Initially, let us consider the problem of representing $N$ $d$-dimensional samples, image vectors $\{\underline{x}_1, \underline{x}_2, \dots, \underline{x}_N\}$ by a single vector $\underline{x}_0$. To be more specific, suppose that we want to find a vector $\underline{x}_0$ such that the sum of the squared distances between $\underline{x}_0$ and the various $\underline{x}_k$, $k = 1\dots,N$, is as small as possible. Let us define a cost function $J_0(\underline{x}_0)$ that must be minimized as:

$$J_0(\underline{x}_0) = \sum_{k=1}^{n} \left\| \underline{x}_0 - \underline{x}_k \right\|^2 . \tag{4.2}$$

Next, we will show that the solution to this problem is given by $\underline{x}_0 = \underline{m}$, where $\underline{m}$ is the dataset sample mean,

$$\underline{m} = \frac{1}{n} \sum_{k=1}^{n} \underline{x}_k . \tag{4.3}$$

First, let us rewrite $J_0(\underline{x}_0)$ as follows,

$$
\begin{aligned}
J_0(\underline{x}_0) &= \sum_{k=1}^{n} \left\| (\underline{x}_0 - \underline{m}) - (\underline{x}_k - \underline{m}) \right\|^2 \\
&= \sum_{k=1}^{n} \left\| \underline{x}_0 - \underline{m} \right\|^2 - 2\sum_{k=1}^{n} (\underline{x}_0 - \underline{m})^t (\underline{x}_k - \underline{m}) + \sum_{k=1}^{n} \left\| \underline{x}_k - \underline{m} \right\|^2 \\
&= \sum_{k=1}^{n} \left\| \underline{x}_0 - \underline{m} \right\|^2 - 2(\underline{x}_0 - \underline{m})^t \sum_{k=1}^{n} (\underline{x}_k - \underline{m}) + \sum_{k=1}^{n} \left\| \underline{x}_k - \underline{m} \right\|^2 \\
&= \sum_{k=1}^{n} \left\| \underline{x}_0 - \underline{m} \right\|^2 + \sum_{k=1}^{n} \left\| \underline{x}_k - \underline{m} \right\|^2 .
\end{aligned}
\tag{4.4}
$$

Note that the above expression is minimized when $\underline{x}_0 = \underline{m}$, since the second term in the summation is independent of $\underline{x}_0$.

Next, let us consider $\underline{e}$ to be a unit vector in the direction of the line running through the sample mean. The equation of the line can be written as:

$$\underline{x} = \underline{m} + a\underline{e}, \tag{4.5}$$

where the scalar $a$ (which can take any real value) corresponds to the distance of any point $\underline{x}$ from the mean $\underline{m}$. At this stage, we can find an "optimal" set of coefficients $a_k$'s, if we represent $\underline{x}_k$ by $\underline{m} + a_k\underline{e}$, by minimizing the following cost function:

$$J_1 = (a_1, a_2, ..., a_n, \underline{e}) = \sum_{k=1}^{n} \left\| (\underline{m} + a_k\underline{e}) - \underline{x}_k \right\|^2 = \sum_{k=1}^{n} \left\| a_k\underline{e} - (\underline{x}_k - \underline{m}) \right\|^2$$

$$= \sum_{k=1}^{n} a_k^{\,2} \left\| \underline{e} \right\|^2 - 2\sum_{k=1}^{n} a_k \underline{e}^{\,t}(\underline{x}_k - \underline{m}) + \sum_{k=1}^{n} \left\| \underline{x}_k - \underline{m} \right\|^2 \tag{4.6}$$

Recognizing that $\|\underline{e}\|$ is equal to 1, partially differentiating with respect to $a_k$ and setting the derivative to zero, leads to:

$$a_k = \underline{e}^{\,t}(\underline{x}_k - \underline{m}). \tag{4.7}$$

Geometrically, this result means that least-square solution may be obtained by projecting the vector $\underline{x}_k$ on the line in the direction of $\underline{e}$ that passes through the sample mean $\underline{m}$.

Next let us consider the problem of finding the best direction $\underline{e}$ for the line of interest. Let us define the total scatter matrix S as:

$$S = \sum_{k=1}^{n} (\underline{x}_k - \underline{m})(\underline{x}_k - \underline{m})^t. \tag{4.8}$$

Replacing the Equation (4.7) into Equation (4.6) leads to:

$$J_1(\underline{e}) = \sum_{k=1}^{n} a_k^{\,2} - 2\sum_{k=1}^{n} a_k^{\,2} + \sum_{k=1}^{n} \left\| \underline{x}_k - \underline{m} \right\|^2$$

$$= -\sum_{k=1}^{n} \left[ \underline{e}^{\,t}(\underline{x}_k - \underline{m}) \right]^2 + \sum_{k=1}^{n} \left\| \underline{x}_k - \underline{m} \right\|^2$$

29

$$= -\sum_{k=1}^{n} \underline{e}^t (\underline{x}_k - \underline{m})(\underline{x}_k - \underline{m})^t \underline{e} + \sum_{k=1}^{n} \left\| \underline{x}_k - \underline{m} \right\|^2$$

$$= -\underline{e}^t S \underline{e} + \sum_{k=1}^{n} \left\| \underline{x}_k - \underline{m} \right\|^2. \qquad (4.9)$$

Naturally, the vector $\underline{e}$ that minimizes $J_1$ also maximizes $\underline{e}^t S \underline{e}$. The solution to this problem must satisfy the equation:

$$S \underline{e} = \lambda \underline{e} . \qquad (4.10)$$

This means that the vector $\underline{e}$ that minimizes $J_1(\underline{e})$ is an eigenvector of the scatter matrix. As a result, maximizing the expression $\underline{e}^t S \underline{e}$ is obtained by selecting for $\underline{e}$ the eigenvector of $S$ associated with the largest eigenvalue. In other words, we project the data onto a line running through the sample mean in the direction of the eigenvector of the scatter matrix having the largest eigenvalue to find the best one-dimensional projection of the data.

Figure 11 presents a two-class, two-dimensional problem where we wish to reduce the data dimension from two to one. Class 1 and 2 are represented by stars and circles respectively. The direction that provides the best one-dimensional PCA representation of the data is the line included in Figure11. Note that class discrimination is preserved as the projected stars and circles onto the line do not overlap.
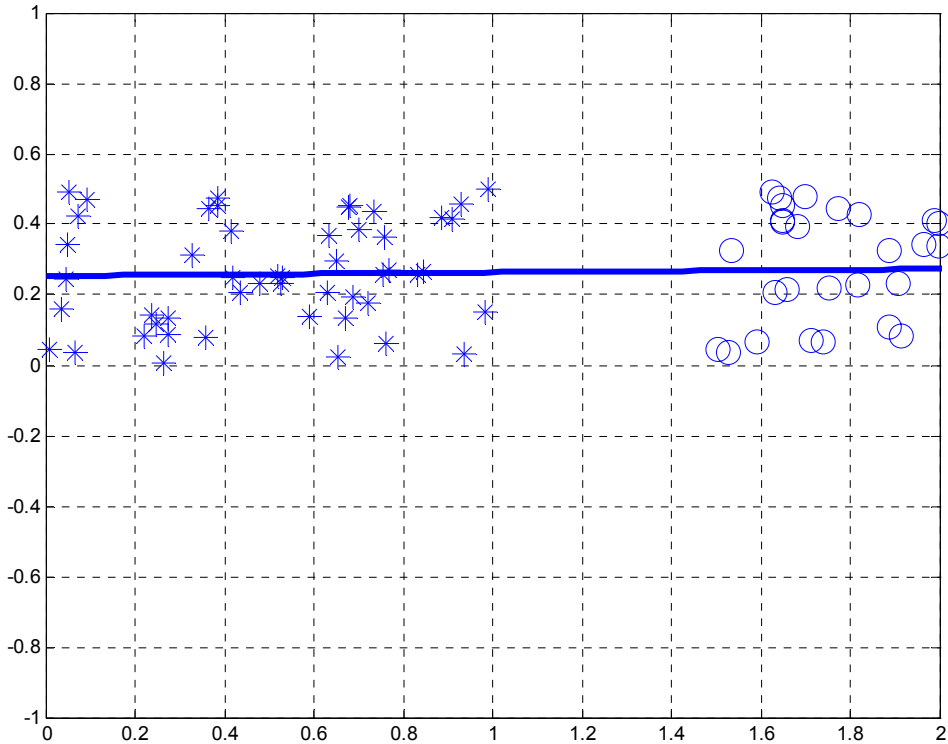
Figure 11.   PCA Example. The line represents the projection direction in which the data should be projected to obtain the best representation in one dimension.

Recall that PCA was initially designed for compression applications, and that the projection directions are not optimized to preserve class separability. Figure 12 shows an example in which the projection direction that best represents the data does not preserve class discrimination when projecting a two-dimensional dataset with two classes onto a one-dimensional space (a line).  In this example, the projected stars and circles overlap, and it is no longer possible to separate the two classes.
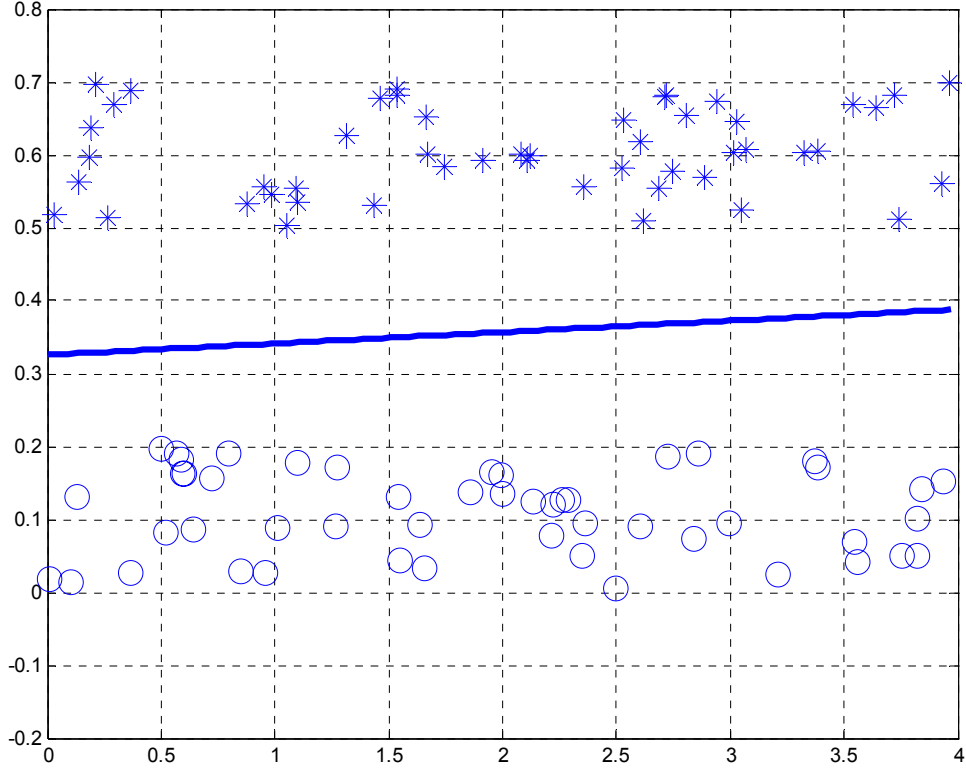
Figure 12.    Data Points and PCA projection. The line provides the best projection direction to represent the data set.

The above result can be extended from a one-dimensional projection to a $p$-dimensional projection.

$$\underline{x} = \underline{m} + \sum_{i=1}^{p} a_i \underline{e}_i \,.$$                    (4.11)

The cost function becomes

$$J_p = \sum_{k=1}^{n} \left\| (\underline{m} + \sum_{i=1}^{p} a_{ki}\underline{e}_i) - \underline{x}_k \right\|^2 = \sum_{k=1}^{n} \left\| \sum_{i=1}^{p} a_{ki}\underline{e}_i - (\underline{x}_k - \underline{m}) \right\|^2 .$$    (4.12)

This function is minimized when the vectors $\underline{e}_1$, $\underline{e}_2,...\underline{e}_p$ are the $p$ scatter matrix eigenvectors associated with the $p$ largest eigenvalues. Note that these eigenvectors are orthonormal as the scatter matrix is real and symmetric, and they form a basis set used to represent any feature vector $\underline{x}$. The coefficients $a_k$ are the projections of $\underline{x}$ in that basis and are called principal components.

32

The PCA scheme may be implemented in a matrix form, as follows:

Each image contained in the training set is reshaped as a column vector with dimension $q$, and the data matrix $A$ is defined as the concatenation of all image vectors columnwise, resulting in a matrix of dimension $q$ (in our case 2700). Next, the mean image of the training set is subtracted from each image resulting in a matrix $X$. The data covariance matrix $S$ is defined as:

$$S = XX^t. \tag{4.13}$$

Recall that the $d$ eigenvectors of $S$ corresponding to the largest eigenvalues minimize the cost function $J(\cdot)$ previously defined. Further, note that the matrix $S$ has dimension 2700 in our case, which is very large. Computing the eigenvectors of $S = XX^t$ may be significantly decreased by computing them in terms of the eigenvectors of $X^tX$ with the snapshot method which is described next.

### b. Snapshot Method

The singular value decomposition theorem states that any $K$ x $N$ matrix can be decomposed and written as the product of three matrices [32]

$$X = U\Sigma V^{*T}, \tag{4.14}$$

where $U$ is the $K$ x $K$ unitary left singular vectors matrix, containing the singular vectors columnwise, $V$ is the N x N unitary right singular vectors matrix, and $\Sigma$ is the $K$ x $N$ diagonal matrix of nonnegative real singular values. The correlation matrix $XX^t$ can be expressed as:

$$XX^t = (U\Sigma V^t)(V\Sigma^T U^t) = U\Sigma(V^tV)\Sigma^t U^t = U\Sigma\Sigma^t U^t. \tag{4.15}$$

By analogy the product $X^tX$ can be expressed as:

$$X^tX = V\Sigma^t\Sigma V^t. \tag{4.16}$$

Equation (4.15) shows that the matrix $U$ contains the eigenvectors of $XX^t$ and the eigenvalues of $XX^t$ are diagonal elements of the matrix $\Sigma\Sigma^t$, by analogy Equation

(4.16) shows that the matrix $V$ contains the eigenvectors of $X^tX$ and that the eigenvalues of $X^tX$ are diagonal elements of the matrix $\Sigma'\Sigma$.

Next, let us consider the product $XV$,

$$XV = U\Sigma V'V = U\Sigma, \qquad (4.17)$$

where the last step follows because $V$ is unitary. Note that the only useful eigenvectors for our application are those associated with non-zero eigenvalues. Therefore we define $\Sigma^{-1}$ as the inverse of the portion of $\Sigma$ which contains the non-zero diagonal elements of $\Sigma$ only. Therefore, the eigenvectors associated with non-zero eigenvalues are defined:

$$U = XV\Sigma^{-1}. \qquad (4.18)$$

### 2. Linear Discriminant Analysis (LDA)

LDA is an approach designed to reduce a dataset dimensionality while maximizing class separation. As a result, the LDA scheme is optimized for classification problems, where the goal is to preserve class discriminative information [31].

Let us consider initially the case of finding one line that best discriminates the projections of a two-class $d$-dimensional dataset and later we will generalize the equation to a $c$ classes problem.

Assume that we have a set of $n$ $d$-dimensional samples $\underline{x}_1, \underline{x}_2, ...\underline{x}_n$, where $n_1$ samples are in subset (i.e., class) $C_1$, and $n_2$ samples are in subset (i.e., class) $C_2$. If we form a linear combination of the components of $\underline{x}$ we obtain:

$$y = \underline{w}^t\underline{x}, \qquad (4.19)$$

and a corresponding set $y_1, y_2,...,y_n$ divided in two subsets $Y_1$ and $Y_2$.

Geometrically, if $||\underline{w}||$ is equal to one, each $y_i$ is the projection of the corresponding $\underline{x}_i$ onto a line in the direction of $\underline{w}$. Note that the magnitude of $\underline{w}$ is of no real significance because it merely scales $\underline{y}$. A measure of the separation between the projected points is the difference of the sample means.

Let us define $\underline{m}_i$ the $d$-dimensional sample mean as follows:

$$\underline{m}_i = \frac{1}{n_i} \sum_{\underline{x} \in X_i} \underline{x}, \qquad (4.20)$$

where $n_i$ is the number of elements on the set $X_i$.

The sample mean for the projected points is given by

$$q_i = \frac{1}{n_i} \sum_{\underline{y} \in Y_i} y = \frac{1}{n_i} \sum_{\underline{x} \in D_i} \underline{w}^t \underline{x} = \underline{w}^t \underline{m}_i . \qquad (4.21)$$

Note that $q$ (above) is simply the projection of $\underline{m}_i$ on $\underline{w}$.

The distance between the projected means is given by:

$$\left| q_1 - q_2 \right| = \left| \underline{w}^t (\underline{m}_1 - \underline{m}_2) \right|. \qquad (4.22)$$

The above distance may be varied by scaling the vector $\underline{w}$. Thus, a good measure of separation for the projected data requires the definition of this distance with respect to some measure of the standard deviations for each class.

We can define the scatter for projected samples by:

$$\left( L_i \right)^2 = \sum_{y \in Y_i} (y - q_i)^2 . \qquad (4.23)$$

The Fisher Linear discriminant employs the linear transformation $\underline{w}$ such that the cost function:

$$J(\underline{w}) = \frac{\left| q_1 - q_2 \right|^2}{\left( L_1 \right)^2 - \left( L_2 \right)^2}, \qquad (4.24)$$

is maximized. The direction $\underline{w}$ maximizing $J(\cdot)$ leads to the best separation between the two projected sets. To obtain $J(\cdot)$ as a function of $\underline{w}$, we define the scatter matrices $S_i$ and $S_w$ by:

$$S_i = \sum_{x \in X_i} (\underline{x} - \underline{m}_i)(\underline{x} - \underline{m}_i)^t , \qquad (4.25)$$

and

$$S_W = S_1 + S_2 . \tag{4.26}$$

Rewriting $L_i$ from Equation (4.23), the scatter from the projected data, we have:

$$\left( L_i \right)^2 = \sum_{x \in X_i} (\underline{w}^t \underline{x} - \underline{w}^t \underline{m}_i)^2 = \sum_{x \in X_i} \underline{w}^t (\underline{x} - \underline{m}_i)(\underline{x} - \underline{m}_i)^t \underline{w}$$

$$= \underline{w}^t S_i \underline{w}. \tag{4.27}$$

Therefore the sum of these scatter matrices can be written

$$\left( L_1 \right)^2 + \left( L_2 \right)^2 = \underline{w}^t S_W \underline{w} . \tag{4.28}$$

Similarly, the separations of the projected means is given by:

$$\left( q_1 - q_2 \right)^2 = \left( \underline{w}^t \underline{m}_1 - \underline{w}^t \underline{m}_2 \right)^2 = \underline{w}^t \left( \underline{m}_1 - \underline{m}_2 \right) \left( \underline{m}_1 - \underline{m}_2 \right)^t \underline{w}$$

$$= \underline{w}^t S_B \underline{w}, \tag{4.29}$$

where $S_B = (\underline{m}_1 - \underline{m}_2)(\underline{m}_1 - \underline{m}_2)^t . \tag{4.30}$

$S_w$ is the within-class scatter matrix. It is symmetric and positive definite, and it is usually nonsingular if $n > d$. Likewise, $S_B$ is called the between-class scatter matrix. It is also symmetric and positive definite, and has rank one  because it is defined as the outer product of two vectors. In particular, for any $\underline{w}$, $S_b \underline{w}$ lies in the direction of $\underline{m}_1 - \underline{m}_2$.

The cost function $J(\cdot)$ can be expressed  in terms of $S_B$ and $S_W$  as:

$$J(\underline{w}) = \frac{\underline{w}^t S_B \underline{w}}{\underline{w}^t S_w \underline{w}} . \tag{4.31}$$

Therefore,  the vector $\underline{w}$ that maximizes $J(\cdot)$  must satisfy $S_B \underline{w} = \lambda S_W \underline{w}$, which is a generalized eigenvalues problem. This generalized eigenvalue problem may be reformulated as conventional eigenvalue problem when $S_W$ is nonsingular, writing

$$S_W^{-1} S_B \underline{w} = \lambda \underline{w} . \tag{4.32}$$

In our particular case, it is unnecessary to solve for the eigenvalues and eigenvectors of $S_W^{-1} S_B$ due to the fact that $S_B \underline{w}$ is always in the direction of $\underline{m}_1 - \underline{m}_2$. As a result, vector $\underline{w}$ that optimizes $J(\cdot)$ is given by:

$$w = S_W^{-1} \left( \underline{m}_1 - \underline{m}_2 \right). \tag{4.33}$$

As an example consider the two-dimensional two-class data points shown in Figure 13, where we are interested in the one-dimensional projection which preserves the class separation. The vector $\underline{w}$ that maximizes the discriminant between the two classes (stars and circles) is the vertical line which shows the projected data can still be discriminated on the line.
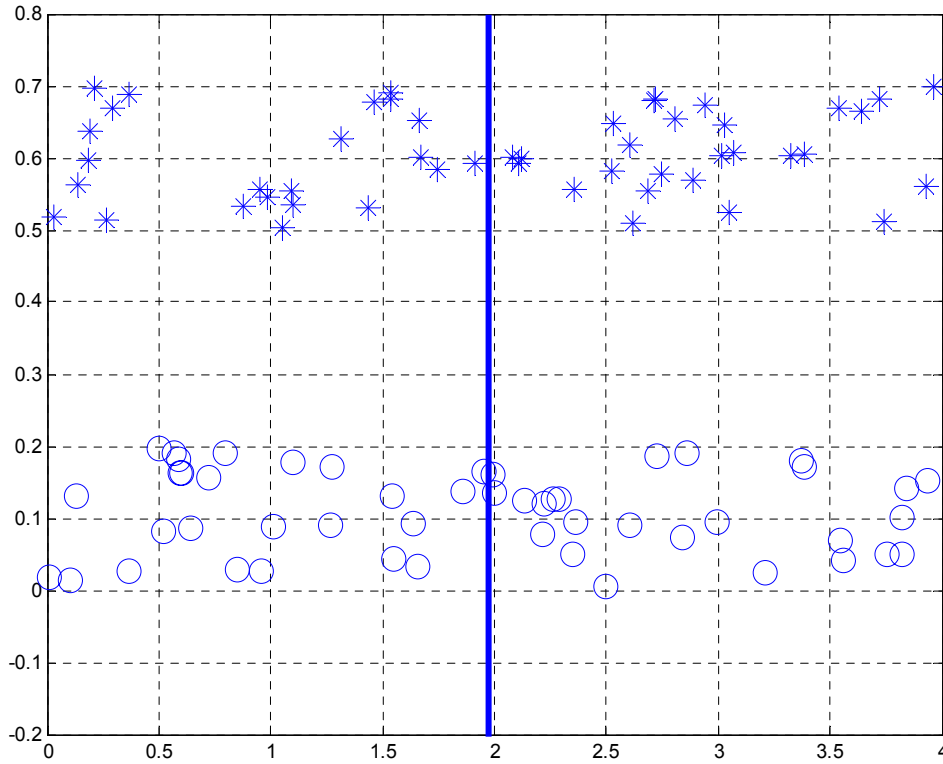


Figure 13. Data points and LDA. The line provides the best projection direction to discriminate the data sets.

The linear discriminant can be extended to a $c$-class problem. The natural generalization involves $c-1$ discriminant functions. Thus, the projection is from a $d$-dimensional space to a $(c-1)$ dimensional space, where it is assumed that $d \geq c$.

The within-class scatter matrix for the $c$-class problem is defined as:

$$S_W = \sum_{i=1}^{c} S_i , \tag{4.34}$$

where,

$$S_i = \sum_{x \in X_i} (\underline{x} - \underline{m}_i)(\underline{x} - \underline{m}_i)^t \tag{4.35}$$

and

$$m_i = \sum_{\underline{x} \in X_i} \underline{x} . \tag{4.36}$$

The proper generalization for $S_B$ is not quite so obvious. Suppose that we define a total mean vector $\underline{m}$ and a total scatter matrix $S_T$ by:

$$\underline{m} = \frac{1}{n} \sum_{x \in \; all \; X_i} \underline{x} , \tag{4.37}$$

and

$$S_T = \sum_{x} (\underline{x} - \underline{m})(\underline{x} - \underline{m})^t . \tag{4.38}$$

Then it follows that

$$
\begin{aligned}
S_T &= \sum_{i=1}^{c} \sum_{x \in X_i} (\underline{x} - \underline{m}_i + \underline{m}_i - \underline{m})(\underline{x} - \underline{m}_i + \underline{m}_i - \underline{m})^t \\
&= \sum_{i=1}^{c} \sum_{x \in X_i} (\underline{x} - \underline{m}_i)(\underline{x} - \underline{m}_i)^t + \sum_{i=1}^{c} \sum_{x \in X_i} (\underline{m} - \underline{m}_i)(\underline{m} - \underline{m}_i)^t \\
&= S_W + \sum_{i=1}^{c} n_i (\underline{m}_i - \underline{m})(\underline{m} - \underline{m}_i)^t .
\end{aligned} \tag{4.39}
$$

The second term in Equation (3.39) is the general between-class scatter matrix, so that the total scatter is the sum of the within-class scatter and the between-class scatter.

The projection from a $d$-dimensional space to a $(c-1)$ dimensional space is accomplished by $c-1$ discriminant functions as: $y_i = \underline{w}_i^t \underline{x}$ , $i = 1,\ldots,\ c-1$. Note that the parameters $y_i$ may be viewed as the components of a vector $\underline{y}$ and the weight vectors $w_i$ may be viewed as the columns of a d-by-c matrix $W$. In such a case the projection operation can be rewritten as a single matrix equation:

$$\underline{y} = W^t \underline{x} . \tag{4.40}$$

Thus, the samples $\underline{x}_1,\ \ldots\ ,\ \underline{x}_n$ project to a corresponding set of samples $\underline{y}_1,\ldots,\underline{y}_n$ which can be described by their own mean vectors and scatter matrices.

Let us define.

$$\underline{q}_i = \frac{1}{n_i} \sum_{\underline{y} \in Y_i} \underline{y} , \tag{4.41}$$

$$\underline{q} = \frac{1}{n} \sum_{\underline{y} \in all Y_i} \underline{y} , \tag{4.42}$$

$$L_W = \sum_{i=1}^{c} \sum_{\underline{y} \in Y_i} (\underline{y} - \underline{q}_i)(\underline{y} - \underline{q}_i)^t , \tag{4.43}$$

and

$$L_B = \sum_{i=1}^{c} n_i (\underline{q}_i - \underline{q})(\underline{q}_i - \underline{q})^t . \tag{4.44}$$

It can be proved [31] that

$$L_W = W^t S_W W , \tag{4.45}$$

and

$$L_B = W^t S_B W . \tag{4.46}$$

At this point, we seek is a transformation matrix $W$ that in some sense maximizes the ratio of the between-class scatter to the within-class scatter. A simple scalar measure of the scatter is the determinant of the scatter matrix. Recall that the determinant is the

product of the eigenvalues and, hence, is the product of the "variances" in the principal direction.

Using this measure, we obtain the cost function:

$$J(W) = \frac{|L_B|}{|L_W|} = \frac{|W^t S_B W|}{|W^t S_W W|}.$$  (4.47)

The columns of an optimal $W$, i.e., defined as maximizing the cost function $J(\cdot)$, corresponds to the largest eigenvalues of the generalized eigenproblem:

$$S_B \underline{w}_i = \lambda_i S_W \underline{w}_i.$$  (4.48)

Note that the within-class scatter matrix $S_W$ is always singular in the face recognition problem. This singularity constraint is due to the fact that the rank of $S_W$ is at most $N - c$, where $N$ is the number of images, $c$ is the number of classes and the number of images is smaller than the number of pixel in each image $n$.

In order to overcome this complication of a singular $S_W$, an alternate cost function proposed by Belhumeur, Hespanha and Kriegman [33] was used in this work. This method, called Fisherface, avoids the singularity problem by projecting the image set to a lower dimensional space so that the resulting within-class scatter matrix $Sw$ is nonsingular. The dimension reduction is achieved by using PCA to reduce the feature space dimension to $N - c$, where N it the total number of images used on the training set, and then applying the standard LDA to further reduce the dimension to $c - 1$.

### 3. Minimum Distance Classifier

Once projection directions are computed from the training set, class-specific centroids are obtained by computing the average values of the projected data for each class. Next the testing images are projected using the projection directions defined for the training set, and the distance between this projected vector and each class centroids is computed. The classification decision is obtained by selecting the class that is the closest in norm to the projected testing image.

This Chapter presented the two linear dimension reduction schemes implemented, the snapshot method implemented to reduce the computational cost associated with the PCA-based algorithms, and described the minimum distance classifier implemented in this study. Next Chapter presents the classification results obtained with our IR database.

THIS PAGE INTENTIONALLY LEFT BLANK

# V. RESULTS

PCA and LDA-based face recognition system classification performances are estimated using a cross-validation variant to take into account the relatively small size of the database (420 images corresponding to 14 individuals with 30 images each were generated and used in the results).

## A. CROSS VALIDATION

Cross-validation is a statistical scheme designed to estimate generalization errors based on "resampling". It is quite useful when dealing with small databases as it averages classification performances over multiple testing subsets [34]. The basic $k$-fold cross-validation implementation divides the dataset into $k$ disjoint sets, where training is conducted with $k-1$ datasets and tested with the remaining set. The process is repeated $k$ times, where each time the selected testing set is different, and the overall classification performance is obtained as the mean of all performances found.

In this work, we consider a cross-validation variant, where the data available is separated in two disjoint sets, the training and the validation set, as shown in Figure 14. The training set containing 60% of the pictures available is used to obtain the projection directions and class centroids. The validation set contains the remaining 40% of the pictures available and is used to test the system performance. This data split and classification estimation process is repeated 1000 times, while the number of images per subject is kept constant for all subjects, and the overall classification performance is computed as the mean of all repetitions.
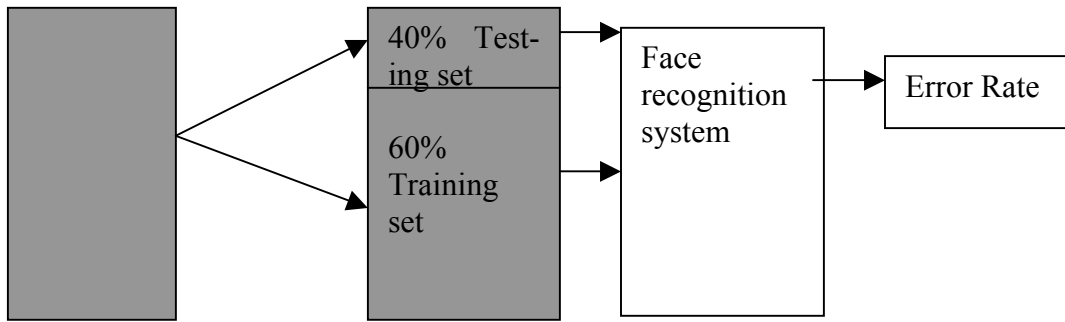
Figure 14.    Block diagrams illustrating the cross validation procedure.

## B. PCA RESULTS

In addition to the basic PCA-based implementations, we also considered the impact of removing the first few eigenvector contributions, as is done successfully in visible imagery classification to make the recognition algorithms more robust to illumination changes [33].

Several PCA-based variants were investigated:

- PCA100: PCA scheme using the first top 100 eigenvectors for projection directions (i.e., the eigenvectors associated with the 100 largest eigenvalues);

- PCA40: PCA scheme using the first top 40 eigenvectors for projection directions;

- PCA40W1: PCA scheme using the first 40 eigenvectors, after removing the top first eigenvector;

- PCA40W2: PCA scheme using the first 40 eigenvectors, after removing the top two eigenvectors;

- PCA40W3: PCA scheme using the first 40 eigenvectors, after removing the top three eigenvectors;

- PCA40W4: PCA scheme using the first 40 eigenvectors, after removing the top four eigenvectors;

- PCA40W5: PCA scheme using the first 40 eigenvectors, after removing the top five eigenvectors.

The maximum number of PCA-based projection directions is related to the dimension of the data used, but some projection directions may be disregarded because the corresponding eigenvalues associated are close to zero. It can be shown that the maximum number of PCA-based projection directions associated with non-zero eigenvalues is equal to the size of the dataset when the database size is smaller than the data dimension. In our work the maximum number of PCA-based projection directions is 2700, and the number of PCA-based projection directions associated with non-zero eigenvalues is 256 (that is, equal to 60% of 420). The number of "useful" PCA-based projection direction was found experimentally to be equal 100. Note that we defined a projection direction (i.e., eigenvector) as "useful" when the associated eigenvalue is at least equal to 0.1% of the largest value. Figure 15 plots the overall classification performance when a combination of datasets #1 and #6 were used as the training set, and dataset #5 was selected for testing. Figure 15 shows that the best classification performance that can be achieved with the data used in this study is obtained with the top thirty-one eigenvectors (leading to an error rate of 26.5%). Note however, that this result is for one training/testing split combination. Similar behavior was observed for other combinations. Therefore, we selected the top 40 eigenvectors to insure the best obtainable classification performances for all training/testing sets combinations used for the cross-validation implementation.

The cross-validation implementation uses 1000 repetitions, where the datasets are kept identical for all methods used in each case. Classification error histograms for all schemes are shown in Figures 16 to 24. Figure 25 illustrates the LDA classification scheme robustness, as explained further below. Table 2 lists the mean classification error rate in percent for each subject considering the 1000 repetitions.
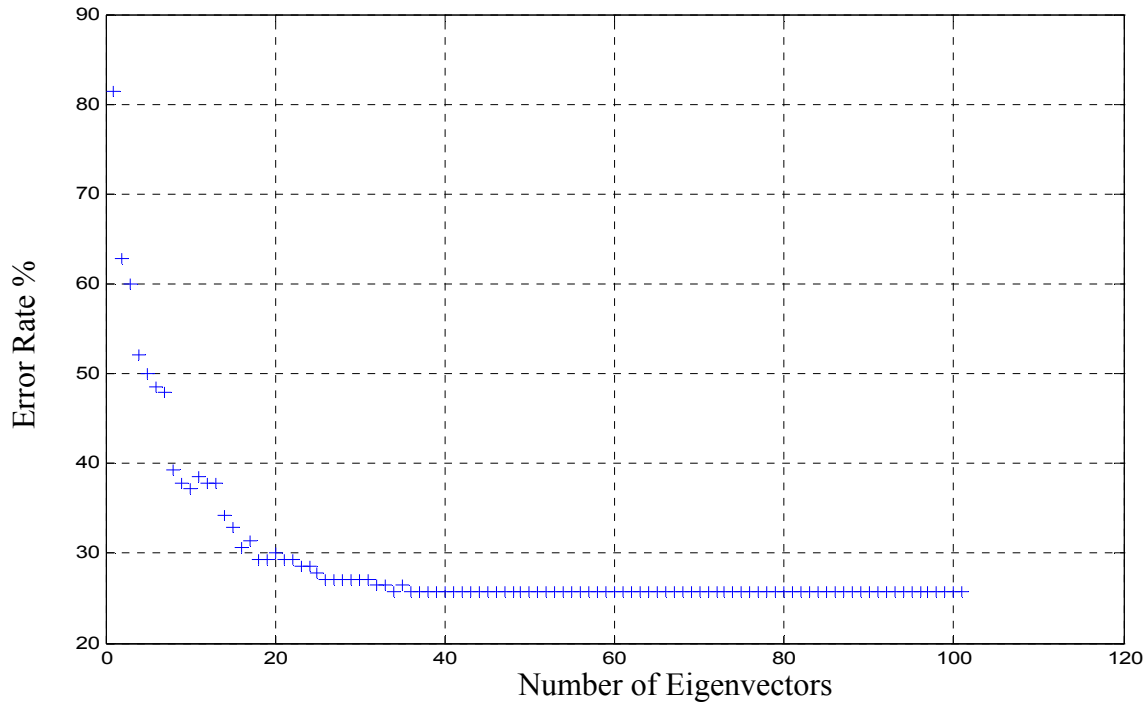
Figure 15. Error rate (%) versus the number of eigenvectors
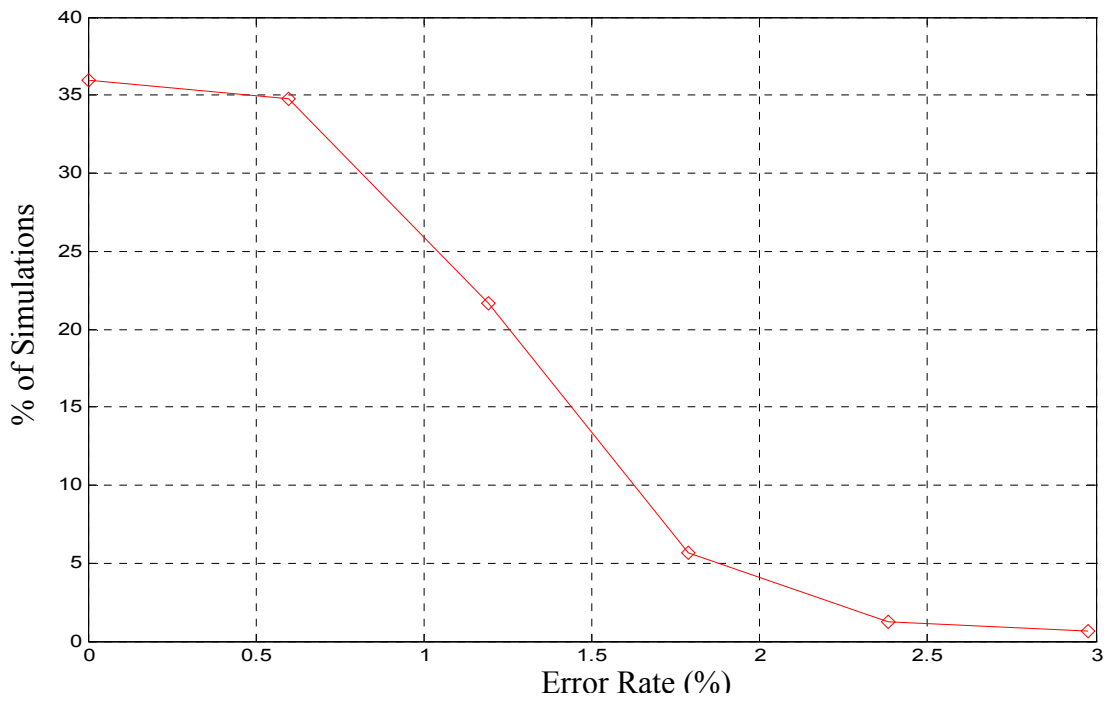


Figure 16. LDA error rate histogram in %; cross-validation implemented
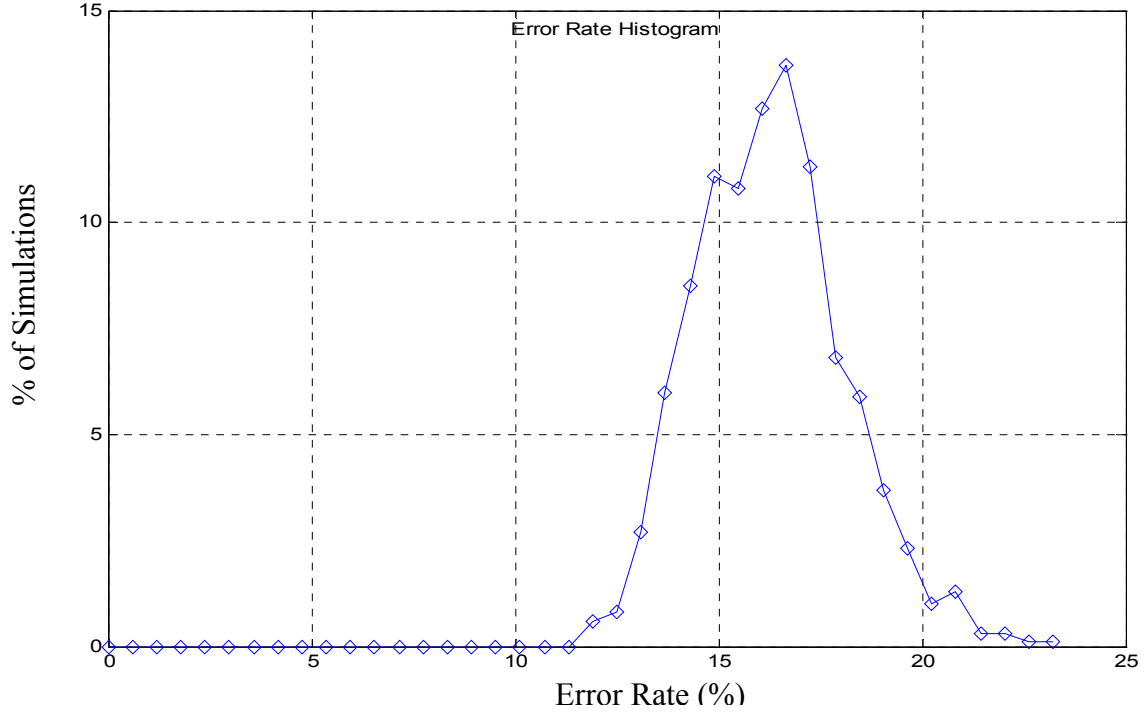
Figure 17.    Direct classification error rate histogram in %; cross-validation implemented.
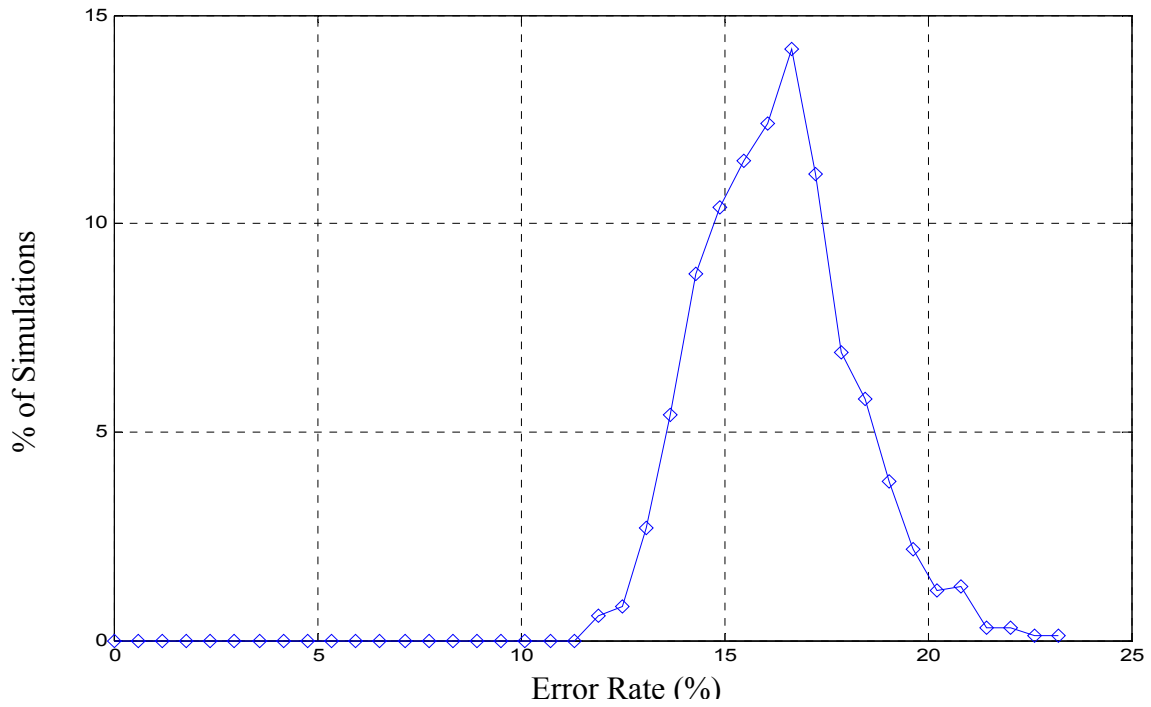


Figure 18.    PCA100 error rate histogram in %; cross validation implemented.
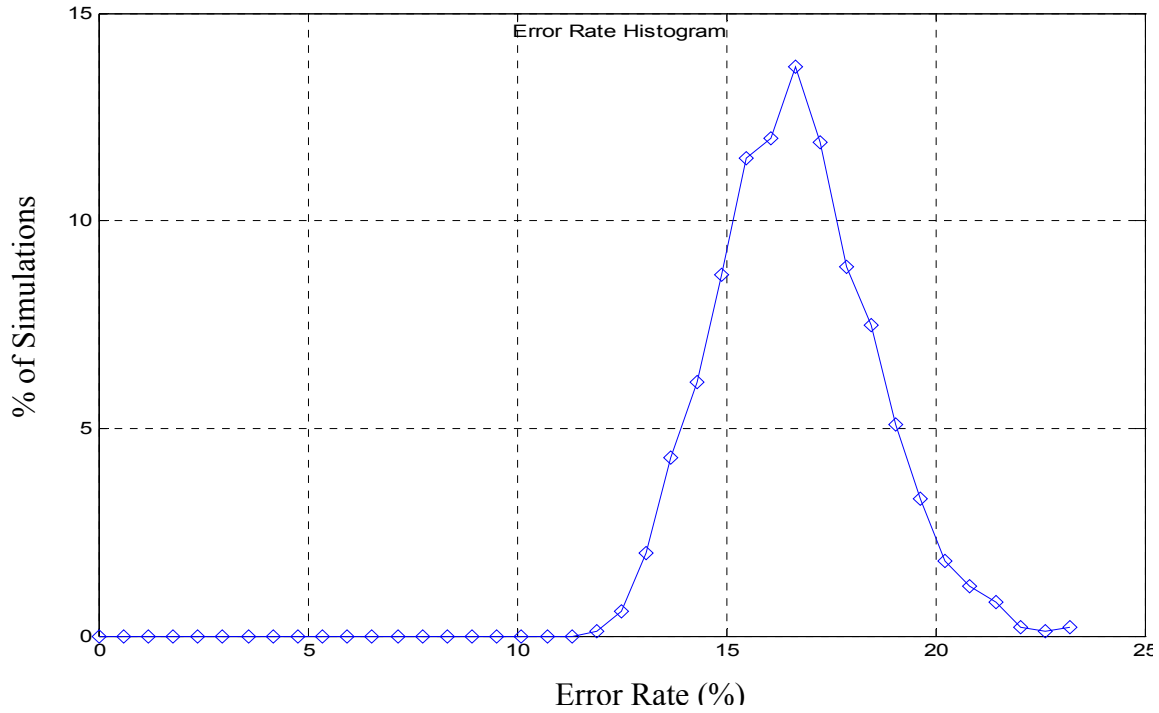
Figure 19.    PCA40 error rate histogram in %; cross validation implemented.
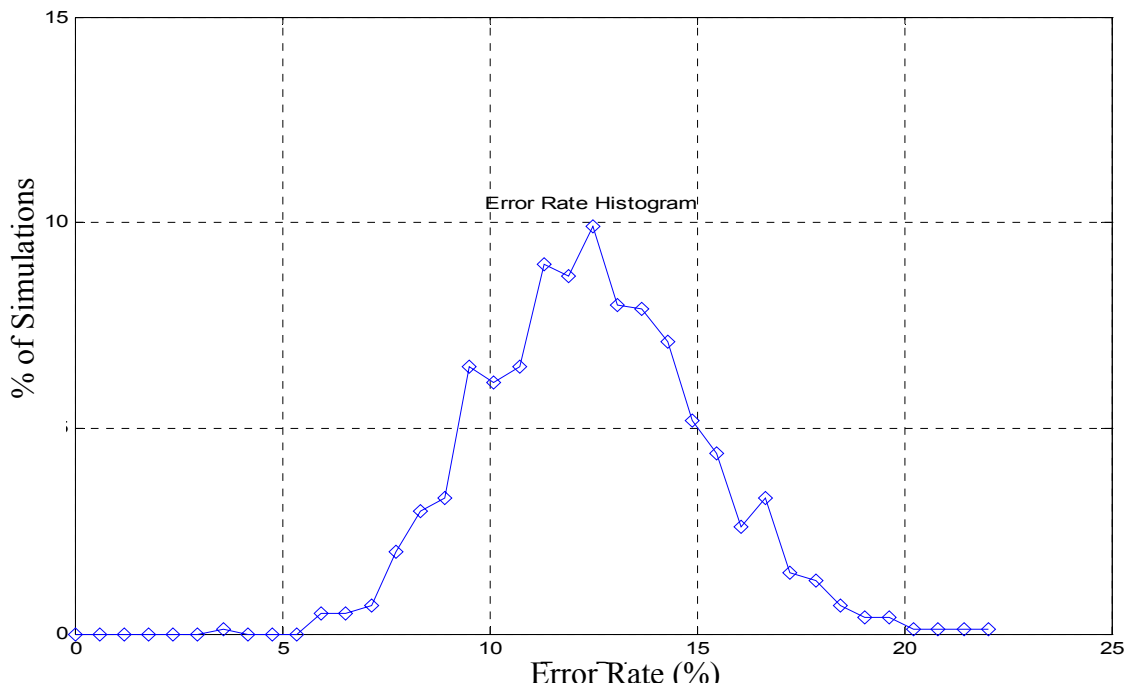


Figure 20.    PCA40W1 error rate histogram in % with first eigenvector removed; cross valida-

tion implemented.
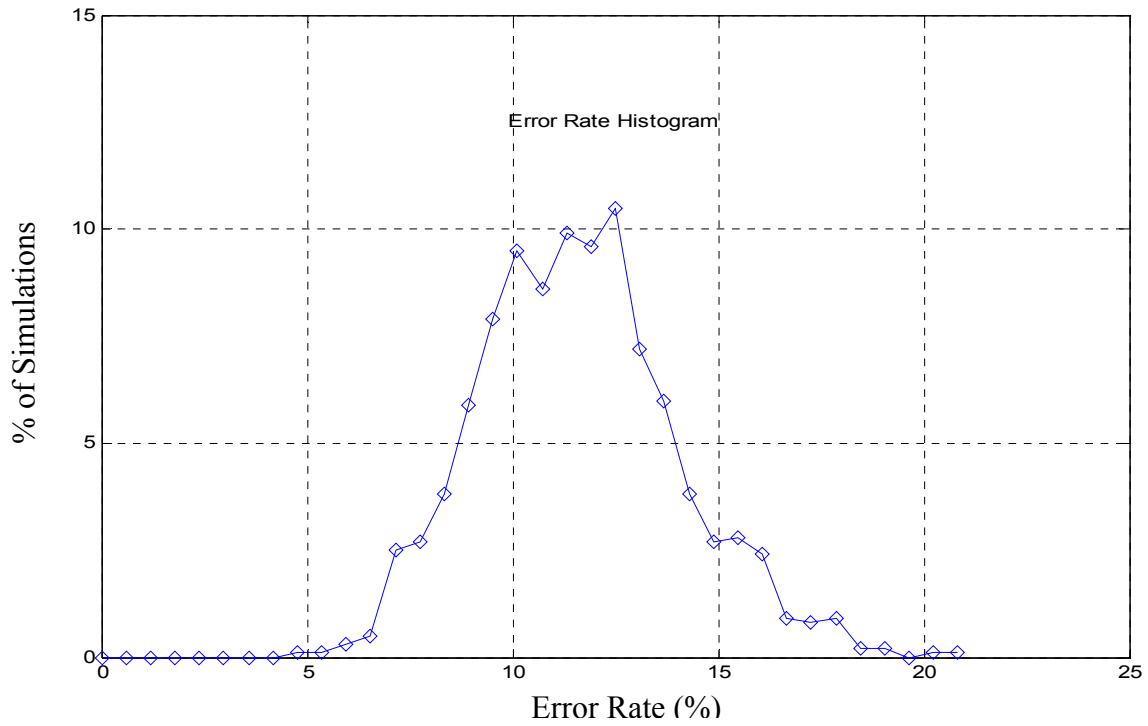
Figure 21.    PCA40W2 error rate histogram in % with first two eigenvectors removed; cross validation implemented.
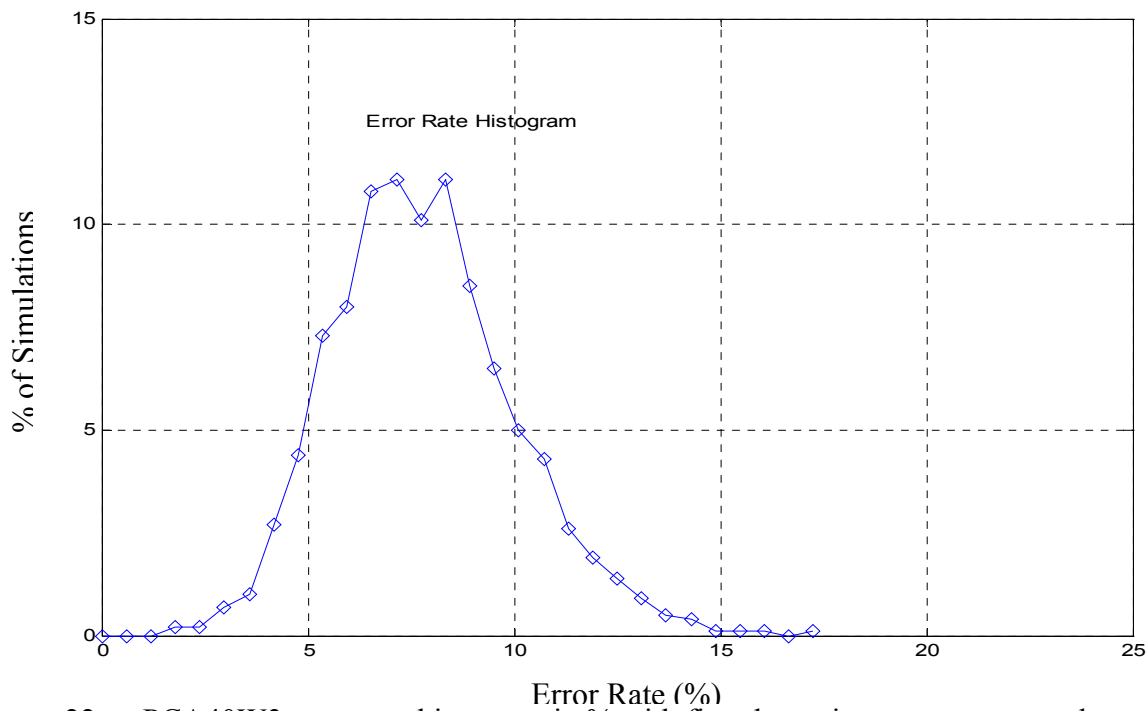


Figure 22.    PCA40W3 error rate histogram in % with first three eigenvector removed; cross validation implemented.
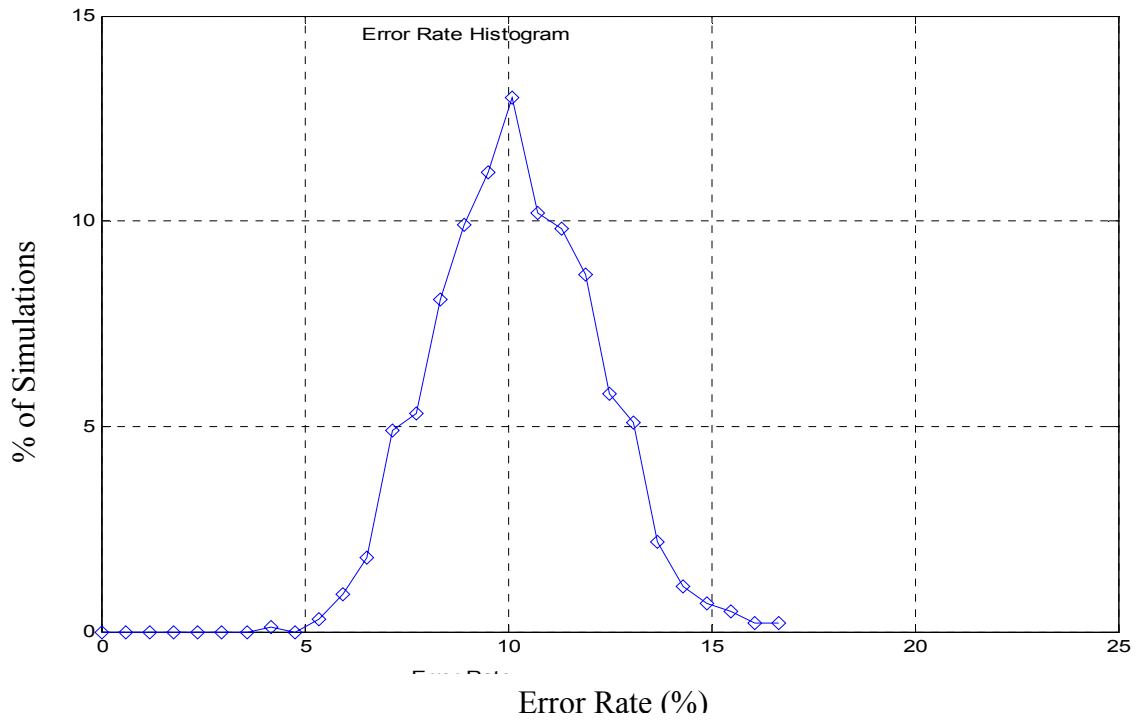
Figure 23.    PCA40W4 error rate histogram in % with first four eigenvector removed; cross validation implemented.
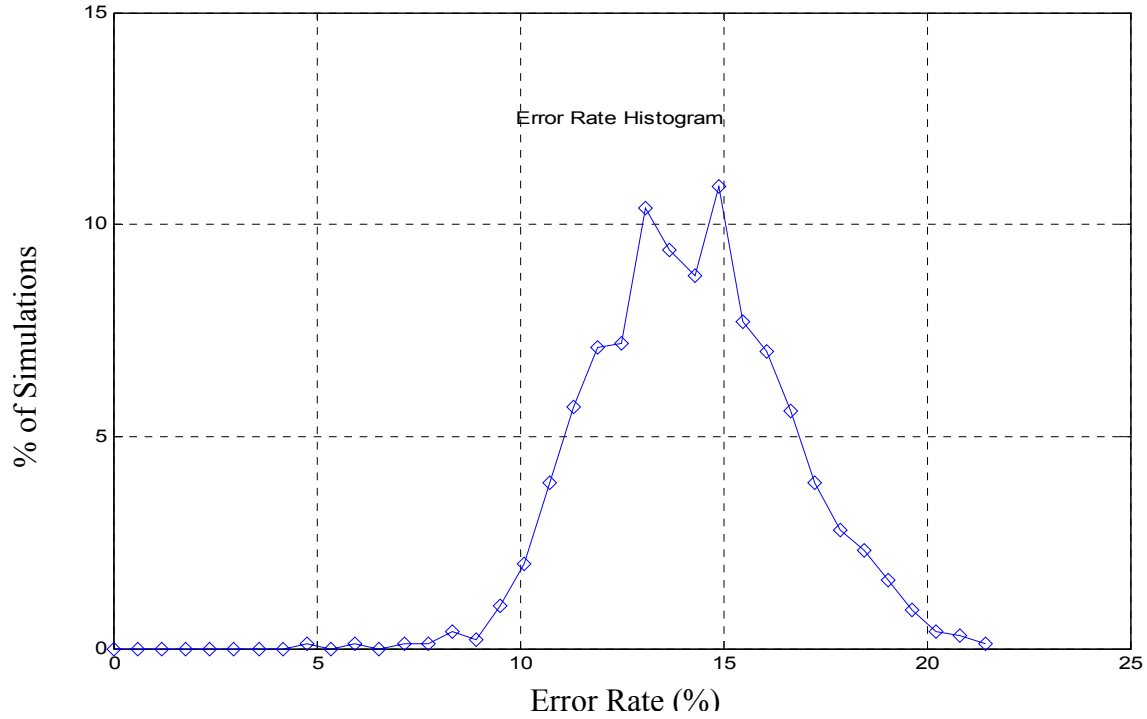
Figure 24.    PCA40W5 error rate histogram in % with the first five eigenvectors removed; cross validation implemented.
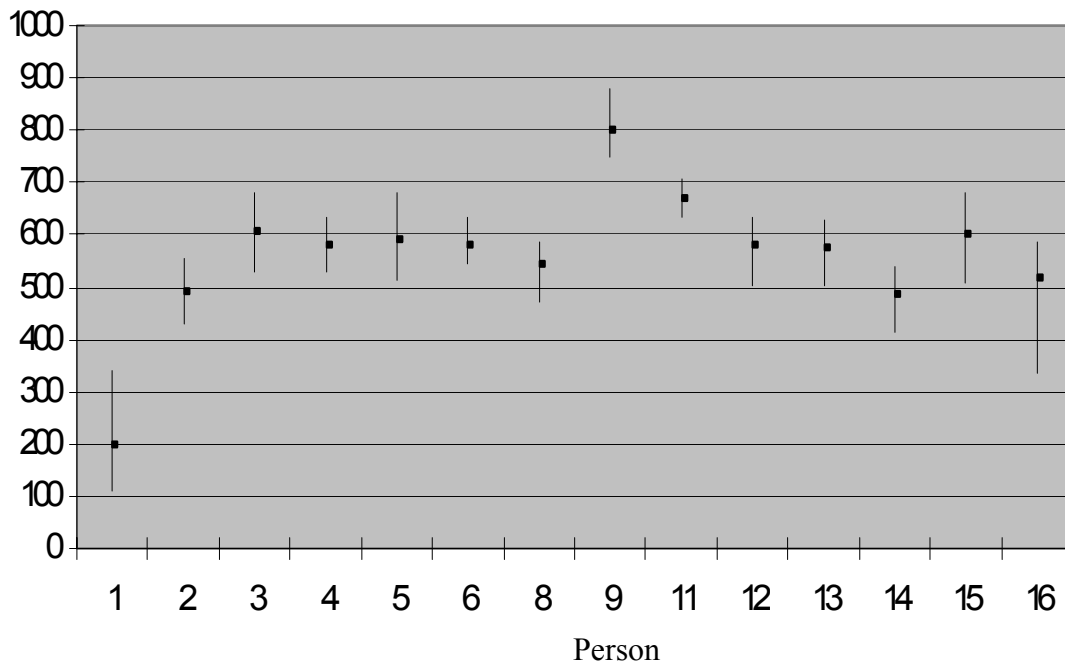


Figure 25.    LDA scheme; Minimum/Mean/Maximum class distances from testing set subject #1 to all training set class-specific centroids; training set: datasets #1 and #5; testing set: dataset #6.

Table 2.    Error rate (%) per class.

| Class → | 1 | 2 | 3 | 4 | 5 | 6 | 8 | 9 | 11 | 12 | 13 | 14 | 15 | 16 | Mean Error Rate (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | | | | | | | | | | | | | | | |
| Direct | 42.62 | 36.08 | 11.08 | 14.07 | 0.01 | 16.01 | 11.93 | 0.01 | 10.05 | 3.34 | 75.72 | 3.29 | 0.19 | 2.76 | 16.22 |
| PCA100 | 42.65 | 36.09 | 11.18 | 14.17 | 0.01 | 16.13 | 11.97 | 0.01 | 10.15 | 3.34 | 75.79 | 3.29 | 0.20 | 2.80 | 16.27 |
| PCA40 | 42.97 | 36.25 | 11.93 | 14.75 | 0.01 | 16.63 | 12.10 | 0.01 | 10.99 | 3.38 | 75.88 | 3.33 | 0.30 | 3.58 | 16.58 |
| PCA40W1 | 19.87 | 7.93 | 26.23 | 0.00 | 30.37 | 11.39 | 22.45 | 0.00 | 8.68 | 3.93 | 30.49 | 0.08 | 9.65 | 5.02 | 12.58 |
| PCA40W2 | 15.12 | 11.24 | 13.03 | 0.00 | 30.03 | 12.85 | 23.06 | 0.04 | 10.53 | 3.32 | 33.84 | 0.76 | 6.39 | 3.69 | 11.71 |
| PCA40W3 | 15.11 | 9.06 | 7.51 | 0.00 | 17.40 | 3.21 | 15.98 | 0.04 | 5.11 | 3.93 | 24.31 | 0.26 | 1.99 | 5.08 | 7.78 |
| PCA40W4 | 22.18 | 10.00 | 22.52 | 0.00 | 11.40 | 5.08 | 14.93 | 3.11 | 3.92 | 4.78 | 32.56 | 0.54 | 0.05 | 11.13 | 10.16 |
| PCA40W5 | 23.52 | 12.72 | 33.63 | 0.07 | 22.39 | 8.60 | 20.59 | 21.48 | 6.30 | 10.57 | 21.21 | 3.08 | 0.48 | 13.16 | 14.13 |
| LDA | 2.50 | 0.00 | 1.91 | 0.00 | 0.03 | 1.75 | 0.30 | 0.02 | 0.50 | 0.00 | 0.83 | 0.00 | 0.12 | 0.50 | 0.60 |

The following comments can be made:

1. Figure 16 shows the results for the LDA scheme, it provides an overall average classification error rate equal to 0.62%. On 36% of the repetitions there were no classifications errors and on 35% of the simulations there was 1 classification error.

2. Figure 17 shows the results for the direct classification without the use of dimension reduction. On Figures 18 and 19 are depicted the results for the PCA100 and PCA40 schemes. From these figures we verify that there were no degradation on the system performance as we reduce the dimension of the data before applying the minimal distance classifier.

3. From Figures 20, 21 and 22 we verify that the mean error rate decreases as we remove the first top three eigenvectors. From Figures 23 and 24 we verify that the mean error rate increases if we eliminate further top eigenvectors. The best PCA-based scheme was the PCAW3 that has an overall classification error rate equal to 7.78%.

4. Figure 25 illustrates the LDA-based classification performance robustness by plotting the distances between one image of test subject #1 to all 16 training set centroids, when datasets #1 and #5 are selected for training and dataset #6 is used for testing. The bar plot shows minimum, maximum and average distances from test subject #1 to any class-specific centroid. Note that the distance from test subject #1 to the class #1 centroid is significantly smaller than it is to any other class centroid, indicating the class-specific clusters are well separated as far as the test subject is considered, which makes the classification decision easy to make. Similar behavior can be observed for all other test subjects in this testing set.

5. The LDA-based classification performance was evaluated by removing the largest eigenvector but this step did not result in any improvement.

6. Table 2 shows the performance of each classification method for each set of images (i.e., each "class" of images). The results show the sensitivity of each

53

class of images to the classification method. For example, some classes showed improvement with the removal of the top eigenvector; others did not. From the right column of the table, we see that the LDA method had the lowest mean error rate. The PCA40W3 had the lowest mean error rate of all of the PCA-based methods.

This Chapter presented the face recognition classification performances obtained with the two dimension reduction approaches considered in this study. Results show that best performances are obtained with the Linear Discriminant implementation, as expected from its definition that is best matched for classification. Results also show that the PCA-based implementation benefits from removing the top three eigenvectors, pointing out the fact that there are still brightness issues which need to be addressed when dealing with IR imagery.

# VI. CONCLUSIONS

This study investigated a (IR) face recognition system using an uncooled IR camera. A computer-based image collection set-up was designed and used to collect a small database. The small database has 420 facial images, from 14 volunteers (13 adult males and 1 adult female) with 30 pictures per person. All subjects are without glasses. Three different facial expression sets were collected; first, the individuals wore a neutral expression. The second set was constituted of the same subjects with a "smiling" expression. The third set was collected with individuals pronouncing the vowel "u". Study variables were reduced by using a controlled environment with the same background for each picture, with the person-camera distance fixed, and by restricting the pictures to frontal facial images while allowing a vertical and horizontal angle freedom of $10^{o}$.

Manual and automated facial image cropping routines were implemented, as described earlier in Chapter IV. The cropping format was a simple rectangle of fixed size equal to 60 by 45 pixels. Two linear approaches for the dataset dimension reduction and classification were implemented and their resulting classification performances compared, PCA-based and LDA approaches. A minimum distance classifier was selected to evaluate the classification performances. The overall system performance was evaluated with a cross-validation scheme using 60% of the pictures for training and 40% of the pictures for testing, with 1,000 repetitions.

Results show that the best PCA-based overall classification performance (92.22%) is obtained when selecting the top 40 eigenvectors, while excluding the first three top eigenvectors. The LDA-based approach performed better, with an overall classification performance equal to 99.40%, as expected from the scheme definition.

Results obtained in this study successfully demonstrate that an uncooled IR camera may discriminate between individual subjects obtained from a small database collected under a very controlled environment. The database needs to be extended and more sophisticated image normalization investigated to take into account facial images taken at different distances and/or resolutions. In addition, this study considered two linear pro-

55

jections only. More sophisticated nonlinear implementations are expected to improve the classification performance when dealing with a larger database.

# APPENDIX

This appendix contains all MATLAB-based software implemented for this study.

Images may be loaded into MATLAB individually or in batch mode:

- Individual loading is done with the routine: *readpgm8.m*.

- Batch loading is done with the routine: *LoadImage.m*, where the operator selects the range of images to be loaded in batch mode.

Images may be loaded, cropped and saved back into *.bmp* format with user-specified names in batch mode using the function:

- *CropFaceAuto.m* (calls the automated face location code described below).

Automated face location extraction code is implemented with the functions:

- *locatehead.m* which vertically crops the image around the head,

- *facemapini.m* which computes the three initial face map parameters (by calling the routine *ProjectPCA.m* described below),

- *facemapA.m* which computes the range of possible sub-images contained in the image provided as an input, and leads to the best extracted face (by calling the routine *ProjectPCA.m* described below), FFT implementation.

- *facemap.m* which duplicates the functions *facemapini.m* and *facemapA.m*, without fft implementation (much slower)

Manual face location extraction is implemented with the functions:

- *cropimage.m* which loads the requested image in MATLAB, calls the function *cropface.m* which returns the cropped image, where the cropped is done manually by the operator.

PCA-based dimension reduction schemes is implemented with the function:

- *pca.m* which returns the specified number of projection directions.

57

The LDA-based dimension reduction scheme is implemented with the functions:

- *pca.m* which is first applied to address the singularity issue discussed earlier in Chapter IV,

- *fld.m* which returns projection directions which best discriminate the data.

Class-specific centroid computation is implemented with the function:

- *meanclass.m* which takes the projection direction information obtained with *pca.m* or *fld.m* and computes the centroids of each class selected by the operator.

Classification performance is computed with the function:

- *classify.m* which uses as inputs either the projection direction information or the  images, compares to the class-specific centroids, and returns a class decision.

Cross-validation results are implemented with the two functions:

- *CrossValidateConstant.m* that generates the error rate histograms for a 60% training/40% testing data split, where the number of images per class is kept constant. (Results provided in Chapter V use this routine).

- *CrossValidate.m* that generates the error rate histograms for 60% training 40% testing data split, where the number of images per class is selected randomly and is NOT kept constant.


Working routines:

- *ProjectPCA.m* loads a set of user-specified cropped images, computes the PCA-based projection directions, save all relevant information in the file *face.dat*.  This routine also computes the error rate as a function of the number of eigenvectors, as shown in Figure 15, Chapter V.

- *sortem.m* sorts the eigenvector matrix by decreasing eigenvalue value.

- *InfraredTotalExitance.m* generates the total black body exitance for user-specified wavelength and temperature ranges, used to plot Figure 1, Chapter III.

- *distance.m* computes the distances between two matrices on a column-by-column basis.

- *projectLDA.m* loads cropped image in MATLAB, applies the LDA dimension reduction scheme for a user-specified dataset (training/testing split is user-specified), investigates further dimension reduction, and returns the classification performance. Also offers the option to load the original images.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [W,m,Amean,EVA]=pca(A,n)
% This function computes the pca(Principal Component Analysis) of the data contained
% in the array A.
% input:  A (k x n) contains the data in each column, k = dimension of data vector
%                                               n = number of data samples
% output: W contain n eigenvectors, one in each column
%          m- fraction of the variance in n eigenvectors
%          Amean - mean of the data contained in A in a column
%          Ad - data less the mean Amean, all the data in columns
%          EVA contain the eigenvalues corresponding to the eigenvectors of W on the
%          diagonal
% Obs. The eigenvectors associated with the eigenvalues smaller that 1/1000 of the
% largest eigenvalue are eliminated
% Diogo C. Pereira 1LT BRAF
% 07/26/02
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [W,m,Amean,Ad,EVA]=pca(A,n);
[Adimk,Adimn]=size(A);
if (nargin==1)
   n=Adimn;
end
if (n>Adimn)
   n=Adimn;
end
% Amean is the mean of A using the columns as elements
Amean=mean(A,2);
% Ad is the difference between A and Amean
Ad=A-Amean*ones(1,Adimn);
%eigenvectors (columns of Vectors) and eigenvalues (diag of Values)
[EVE,EVA] = eig(Ad'*Ad);
% obtain index of eigenvalues greater than (0.001 times greater eigenvalue)
[I]=find(EVA>(0.001*max(max(EVA))));
EVAC=zeros(size(EVA));
EVAC(I)=EVA(I);  %EVAC (Eigenvalue Conditioned) will have only values greater
than a minimum value otherwise the value is set to zero
% Obtain n eigenvectors
  %EVAINV is the matrix containing the inverse of the eigenvalues on the diagonal, it
contains zero if the eigenvalue was zero
  % and the inverse of the eigenvalue if the element was larger than zero
  EVAINV=zeros(Adimn); % EVAIN has size n x n since n is the maximum number of
eigenvalues
  EVAINV(I)=1./(sqrt(EVA(I)));
  U=A*EVE*EVAINV;
```

%Sort the vectors/values according with the absolute value of the eigenvalue EVAC and eliminates the columns corresponding
% to zero eigenvalues
[W,EVA]=sortem(U, EVAC);
if (n>size(W,2))
   n=size(W,2);
end
W=W(:,1:n);
temp=diag(EVA);
m=sum(temp(1:n))/sum(temp);
return

---------------------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [W,D]=fld(A,C)
% This function computes the Linear Discriminant W that maximizes
% the ratio |Wt x SB x W|/|Wt x SW x W|  where SW is the within class
% scatter matrix, and SB is the between class scatter matrix
%
% input: A- data matrix, containing data in columns
%         C-row array containing numbers representing the classes of
%         the elements in A
% output: W- weight matrix contains the vectors in columns
%          D - matrix containing the eigenvalues on the diagonal
% last modified: 08/19/2002
% Diogo Pereira
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [W,D]=fld(A,C);
MeanG=mean(A,2);%MeanG is the general mean of the data in A
Ad=A-MeanG*ones(1,size(A,2)); %Ad is matrix A minus the mean of A
ST=Ad*Ad';% ST is the total scatter matrix
clear Ad
Nclass=0; %Nclass counts the number of classes on A
%Compute the variance within all the classes %%%%%%%%%%%%%%%%%%%%%%
SW=zeros(size(A,1)); %SW is the within class scatter matrix
while size(C,2)>0,
   w=C(1,1);
   [I]=find(C==w);
   MeanClass=mean(A(:,I),2);
   Ad=A(:,I)- MeanClass*ones(1,size(A(:,I),2));
   SW=SW+Ad*Ad'; %adds the within class scatter matrix of each class
   A(:,I)=[]; %eliminates the data already used
   C(I)=[];   %eliminates the number corresponding to the class from C vector
   Nclass=Nclass+1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N=min(size(A,1), Nclass-1);%N will be the minimum between the dimension of the data
and Nclass-1

[W,EVA]=eigs(SB,SW,N);

%Set to zero the eigenvalues not equal to a finite value

[I]=find(isfinite(EVA)==1); %Obtain the index of the eigenvalues that are finite

EVAC=zeros(size(EVA));

EVAC(I)=EVA(I);

EVA=EVAC;%EVA will contain just the eigenvalues that are finite

% obtain index of eigenvalues greater than 0.001 times greater eigenvalue

[I]=find(abs(EVA)>(0.001*max(max(abs(EVA)))));

EVAC=zeros(size(EVA));

EVAC(I)=EVA(I); %EVAC will have only values greater than a minimum value other-
wise the value is set to zero

[W,D]=sortem(W,EVAC);% order and eliminates eigenvector corresponding to zero ei-
genvalue

% this part normalizes W so that the norm of each column will be one

NW=ones(size(W,1),1)*sqrt(sum((W.^2),1));

W=W./NW;

return

-------------------------------------------------------------------------------------------------------------------

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% CrossValidateConstant

% This programs test the 2 schemes using cross validation 6 training/4 testing

% The number of elements in each class is left constant

% Diogo Pereira

% 12/14/2002

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

close all;

% Load the images %%%%%%%%%%%%%%%%%%%%%%%%%%

A=[]; % A contains images in columns

T=[]; % T contains the class number of each picture

Crop='a';

N_pictures=10; % number of pictures for each person number

epct=0; % accumulator of the number of errors per class

epct100=0;

epct40=0;

epct40W1=0;

epct40W2=0;

epct40W3=0;

epct40W4=0;

epct40W5=0;

epctLDA=0;

Person=[1 2 3 4 5 6 8 9 11 12 13 14 15 16]; % Person numbers contained on the files

Section=1;

```
[A,T]=LoadImage(A,T,Person, N_pictures,Section,Crop);
Section=5;
[A,T]=LoadImage(A,T,Person, N_pictures,Section,Crop);
Section=6;
[A,T]=LoadImage(A,T,Person, N_pictures,Section,Crop);
% Obtain the samples to be used on the testing and remove it from the training images
% A will contain the training images
% B will contain the testing images
% T is the class number of the images on A
% TB is the class number of the images on B
Atemp=A;
Ttemp=T;
N_trials=300;
N_samples=4*14*3; % we are using 6/4 cross validation

for j=1:N_trials
  A=Atemp;
  T=Ttemp;
  N_Person=14;
  B=[];
  TB=[];
  Remove=[];
  % Generates testing and training sets
  for i=1:(N_Person*3)  % i goes from 1 to the total number of pictures
    f=randperm(10);
    L1=(i-1)*10+1;
    L2=i*10;
    A(:,L1:L2)=A(:,(i-1)*10+f); % mix the samples in each class
    T(:,L1:L2)=T(:,(i-1)*10+f); % adjust the class numbers
    IndSamples=[L1 L1+1 L1+2 L1+3];
    B=[B A(:,IndSamples)];
    TB=[TB T(:,IndSamples)];
    Remove=[Remove IndSamples];
  end
  A(:,Remove)=[];
  T(:,Remove)=[];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PCA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute the pca of the training images A
  [W,m,Amean,Ad]=pca(A);
  % Compute the projection matrix P
  P=W'*(Ad);
  %%%% used to do cross validation directly without pca or lda
  PD=A;
```

```
%AmeanD=mean(A,2);
PD=A-Amean*ones(1,size(A,2));
 % Compute the centroid of each class contained in Person
C=meanclass(P,T,Person);
CD=meanclass(PD,T,Person);
%C(1,:)=[];
 % Using the testing images B
Ad=B-Amean*ones(1,size(B,2));
% Compute the projection matrix P
P=W'*(Ad);
%P(1,:)=[];
% Classify the testing images
dmin=100000000;
 % classifying the images directly
PD=Ad;
[D]=classify(PD,CD,dmin);
NZD=find(D~=0);
D(NZD)=Person(D(NZD));
error=TB-D;
n_errorDirect(j)=sum((error~=0),2);
%computes the number of errors per class
errornz=(error~=0);
epc=reshape(errornz,4,42);
epcp1=epc(:,1:14);
epcp2=epc(:,15:28);
epcp3=epc(:,29:42);
epc=[epcp1;epcp2;epcp3];
epc=sum(epc,1);
epct=epct+epc;
% limiting the size to Neig=number of eigenvectors
% using 100 eigenvectors
Neig=100;
Neig=min(Neig,length(C));
Temp1=P(1:Neig,:);
Temp2=C(1:Neig,:);
[D]=classify(Temp1,Temp2,dmin);
NZD=find(D~=0);
D(NZD)=Person(D(NZD));
error=TB-D;
n_errorPCA100(j)=sum((error~=0),2);
%computes the number of errors per class
errornz=(error~=0);
epc=reshape(errornz,4,42);
epcp1=epc(:,1:14);
epcp2=epc(:,15:28);
epcp3=epc(:,29:42);
```

```
epc=[epcp1;epcp2;epcp3];
epc=sum(epc,1);
epct100=epct100+epc;
%using 40 eigenvectors
Neig=40;
Temp1=P(1:Neig,:);
Temp2=C(1:Neig,:);
[D]=classify(Temp1,Temp2,dmin);
NZD=find(D~=0);
D(NZD)=Person(D(NZD));
%TB
%D40=D
error=TB-D;
n_errorPCA40(j)=sum((error~=0),2);
%computes the number of errors per class
errornz=(error~=0);
epc=reshape(errornz,4,42);
epcp1=epc(:,1:14);
epcp2=epc(:,15:28);
epcp3=epc(:,29:42);
epc=[epcp1;epcp2;epcp3];
epc=sum(epc,1);
epct40=epct40+epc;
%using 40 eigenvectors without the first one
Neig=40;
Temp1=P(2:Neig+1,:);
Temp2=C(2:Neig+1,:);
[D]=classify(Temp1,Temp2,dmin);
NZD=find(D~=0);
D(NZD)=Person(D(NZD));
error=TB-D;
n_errorPCA40W1(j)=sum((error~=0),2);
%computes the number of errors per class
errornz=(error~=0);
epc=reshape(errornz,4,42);
epcp1=epc(:,1:14);
epcp2=epc(:,15:28);
epcp3=epc(:,29:42);
epc=[epcp1;epcp2;epcp3];
epc=sum(epc,1);
epct40W1=epct40W1+epc;
%using 40 eigenvectors without the first two eigenvectors
Neig=40;
Temp1=P(3:Neig+2,:);
Temp2=C(3:Neig+2,:);
[D]=classify(Temp1,Temp2,dmin);
```

```
NZD=find(D~=0);
D(NZD)=Person(D(NZD));
error=TB-D;
n_errorPCA40W2(j)=sum((error~=0),2);
%computes the number of errors per class
errornz=(error~=0);
epc=reshape(errornz,4,42);
epcp1=epc(:,1:14);
epcp2=epc(:,15:28);
epcp3=epc(:,29:42);
epc=[epcp1;epcp2;epcp3];
epc=sum(epc,1);
epct40W2=epct40W2+epc;
%using 40 eigenvectors without the first three eigenvectors
Neig=40;
Temp1=P(4:Neig+3,:);
Temp2=C(4:Neig+3,:);
[D]=classify(Temp1,Temp2,dmin);
NZD=find(D~=0);
D(NZD)=Person(D(NZD));
error=TB-D;
n_errorPCA40W3(j)=sum((error~=0),2);
%computes the number of errors per class
errornz=(error~=0);
epc=reshape(errornz,4,42);
epcp1=epc(:,1:14);
epcp2=epc(:,15:28);
epcp3=epc(:,29:42);
epc=[epcp1;epcp2;epcp3];
epc=sum(epc,1);
epct40W3=epct40W3+epc;
%using 40 eigenvectors without the first four eigenvectors
Neig=40;
Temp1=P(5:Neig+4,:);
Temp2=C(5:Neig+4,:);
[D]=classify(Temp1,Temp2,dmin);
NZD=find(D~=0);
D(NZD)=Person(D(NZD));
error=TB-D;
n_errorPCA40W4(j)=sum((error~=0),2);
%computes the number of errors per class
errornz=(error~=0);
epc=reshape(errornz,4,42);
epcp1=epc(:,1:14);
epcp2=epc(:,15:28);
epcp3=epc(:,29:42);
```

```
epc=[epcp1;epcp2;epcp3];
epc=sum(epc,1);
epct40W4=epct40W4+epc;
%using 40 eigenvectors without the first five eigenvectors
Neig=40;
Temp1=P(6:Neig+5,:);
Temp2=C(6:Neig+5,:);
[D]=classify(Temp1,Temp2,dmin);
NZD=find(D~=0);
D(NZD)=Person(D(NZD));
error=TB-D;
%D40W5=D
n_errorPCA40W5(j)=sum((error~=0),2);
%computes the number of errors per class
errornz=(error~=0);
epc=reshape(errornz,4,42);
epcp1=epc(:,1:14);
epcp2=epc(:,15:28);
epcp3=epc(:,29:42);
epc=[epcp1;epcp2;epcp3];
epc=sum(epc,1);
epct40W5=epct40W5+epc;
%%%%%%%%%%%%%%%%%%%%%%%
% LDA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute the pca of the training images
temp=(N_pictures-1)*length(Person);
[W,m,Amean,Ad,EVA]=pca(A,temp); % This reduces the dimension to N-c = 36 - 6=30
the best result was with 29
% Compute the projection matrix P
P=W'*(Ad);
%   P(1,:)=[];
% Compute the projection matrix P obtained with the FLD(Fisher Linear Discriminant)
[Wopt,D]=fld(P,T);
P=Wopt'*P;
% Computing the centroid of each class
C=meanclass(P,T,Person);
% Using the testing data
Ad=B-Amean*ones(1,size(B,2));
P=W'*(Ad);
%P(1,:)=[];
P=Wopt'*P;
% Classify the testing images
dmin=100000000;
[D]=classify(P,C,dmin);
NZD=find(D~=0);
```

```
 D(NZD)=Person(D(NZD));
 error=TB-D;
 n_errorLDA(j)=sum((error~=0),2);
 %computes the number of errors per class
 errornz=(error~=0);
 epc=reshape(errornz,4,42);
 epcp1=epc(:,1:14);
 epcp2=epc(:,15:28);
 epcp3=epc(:,29:42);
 epc=[epcp1;epcp2;epcp3];
 epc=sum(epc,1);
 epctLDA=epctLDA+epc;
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 end
 h=figure;
 L1=min(n_errorDirect);
 L2=max(n_errorDirect);
 temp1=linspace(0,L2,L2+1);
 N1=hist(n_errorDirect,temp1);
 plot(temp1/N_samples*100,N1/N_trials*100,'bd-');
 title('Histogram of the number of errors');
 xlabel('Percentage of errors');
 ylabel('Percentage of the simulations');
 grid;
 saveas(h,'CrossValidationDirect6_4.fig');
 h=figure;
 L1=min(n_errorPCA100);
 L2=max(n_errorPCA100);
 temp1=linspace(0,L2,L2+1);
 N1=hist(n_errorPCA100,temp1);
 plot(temp1/N_samples*100,N1/N_trials*100,'bd-');
 title('Histogram of the number of errors');
 xlabel('Percentage of errors');
 ylabel('Percentage of the simulations');
 grid;
 saveas(h,'CrossValidation100eig_6_4.fig');
 h=figure;
 L1=min(n_errorPCA40);
 L2=max(n_errorPCA40);
 temp1=linspace(0,L2,L2+1);
 N1=hist(n_errorPCA40,temp1);
 plot(temp1/N_samples*100,N1/N_trials*100,'bd-');
 title('Histogram of the number of errors');
 xlabel('Percentage of errors');
 ylabel('Percentage of the simulations');
 grid;
```

```
saveas(h,'CrossValidation40eig_6_4.fig');
h=figure;
L1=min(n_errorPCA40W1);
L2=max(n_errorPCA40W1);
temp1=linspace(0,L2,L2+1);
N1=hist(n_errorPCA40W1,temp1);
plot(temp1/N_samples*100,N1/N_trials*100,'bd-');
title('Histogram of the number of errors');
xlabel('Percentage of errors');
ylabel('Percentage of the simulations');
grid;
saveas(h,'CrossValidation40W1eig_6_4.fig');
h=figure;
L1=min(n_errorPCA40W2);
L2=max(n_errorPCA40W2);
temp1=linspace(0,L2,L2+1);
N1=hist(n_errorPCA40W2,temp1);
plot(temp1/N_samples*100,N1/N_trials*100,'bd-');
title('Histogram of the number of errors');
xlabel('Percentage of errors');
ylabel('Percentage of the simulations');
grid;
saveas(h,'CrossValidation40W2eig_6_4.fig');
h=figure;
L1=min(n_errorPCA40W3);
L2=max(n_errorPCA40W3);
temp1=linspace(0,L2,L2+1);
N1=hist(n_errorPCA40W3,temp1);
plot(temp1/N_samples*100,N1/N_trials*100,'bd-');
title('Histogram of the number of errors');
xlabel('Percentage of errors');
ylabel('Percentage of the simulations');
grid;
saveas(h,'CrossValidation40W3eig_6_4.fig');
h=figure;
L1=min(n_errorPCA40W4);
L2=max(n_errorPCA40W4);
temp1=linspace(0,L2,L2+1);
N1=hist(n_errorPCA40W4,temp1);
plot(temp1/N_samples*100,N1/N_trials*100,'bd-');
title('Histogram of the number of errors');
xlabel('Percentage of errors');
ylabel('Percentage of the simulations');
grid;
saveas(h,'CrossValidation40W4eig_6_4.fig');
```

```
h=figure;
L1=min(n_errorPCA40W5);
L2=max(n_errorPCA40W5);
temp1=linspace(0,L2,L2+1);
N1=hist(n_errorPCA40W5,temp1);
plot(temp1/N_samples*100,N1/N_trials*100,'bd-');
title('Histogram of the number of errors');
xlabel('Percentage of errors');
ylabel('Percentage of the simulations');
grid;
saveas(h,'CrossValidation40W5eig_6_4.fig');
% hold on
h=figure;
L1=min(n_errorLDA);
L2=max(n_errorLDA);
temp2=linspace(0,L2,L2+1);
N2=hist(n_errorLDA,temp2);
plot(temp2/N_samples*100,N2/N_trials*100,'rd-');
title('Histogram of the number of errors');
xlabel('Percentage of errors');
ylabel('Percentage of the simulations');
grid;
saveas(h,'CrossValidationLDA_6_4.fig');
% saving the error per class
ns=((N_samples/14)*N_trials)/100;
epct=epct/ns;
epct100=epct100/ns;
epct40=epct40/ns;
epct40W1=epct40W1/ns;
epct40W2=epct40W2/ns;
epct40W3=epct40W3/ns;
epct40W4=epct40W4/ns;
epct40W5=epct40W5/ns;
epctLDA=epctLDA/ns;
M=[Person;epct;epct100;epct40;epct40W1;epct40W2;epct40W3;epct40W4;epct40W5;ep
ctLDA];
filename='ErrorPerClass6_4perc';
wk1write(filename,M,4,2);
ns=100/((N_samples/14)*N_trials);
epct=epct/ns;
epct100=epct100/ns;
epct40=epct40/ns;
epct40W1=epct40W1/ns;
epct40W2=epct40W2/ns;
epct40W3=epct40W3/ns;
epct40W4=epct40W4/ns;
```

```
epct40W5=epct40W5/ns;
epctLDA=epctLDA/ns;
M=[Person;epct;epct100;epct40;epct40W1;epct40W2;epct40W3;epct40W4;epct40W5;ep
ctLDA];
filename='ErrorPerClass6_4abs';
wk1write(filename,M,4,2);


%-------------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LocateClassify
% this routine locates a face and classify it according with the PCA method
%
% input:  name of the file containing the image
%         W eigenvectors, H high of the image on the database, C centroids of the classes
%         Amean mean of the data used to obtain the eigenvectors
% output: number of the class the object was classified to
% Obs.: You need to run the program ProjectPCA and ProjectLDA first to obtain
% the files faces and faces2
% Diogo Pereira
% last update: 10/09/2002
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all
clc
% obtaining the name of the image
name=input('Type the name of the file ->','s');
T=3;
B=readpgm8(name); % read pgm files
[HI,WI]=size(B);% obtaining the size of the image
% displays the image%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('Units','pixels','Position',[100 100 WI HI]);
imagesc(B)
set(gca,'Units','pixels','Position',[0 0 WI HI]);
colormap(gray)
pause(.8)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp(['Locating the face, wait...'])
% Locates and obtains the face%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
B=locatehead(B); %locates the head
B=double(B);
load faces W H Amean C
load faces2 W2 C2
[m,n]=size(W);
WI=m/H;
F=facemap(W,H,Amean,B); % returns F the facemap
[M,I]=min(F);
[N,yini]=min(M);
```

```
xini=I(yini);
face=B(xini:(xini+H-1),yini:(yini+WI-1)); % obtain the face on the image
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% displays the face image%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('Units','pixels','Position',[400 100 WI H]);
imagesc(face)
set(gca,'Units','pixels','Position',[0 0 WI H]);
colormap(gray)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% classify the face image%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
face=reshape(face,WI*H,1);
face=face-Amean;
%classify using the PCA scheme
PB=W'*face;
dmin=100000000;
kmax=size(PB,1);
k=kmax;
D=classify(PB(1:k,:),C(1:k,:),dmin);
disp(['You belongs to class ',num2str(D),' using the PCA classification scheme'])
%classify using the LDA scheme
PC=(W2)'*face;
dmin=100000000;
kmax=size(PC,1);
k=kmax;
D=classify(PC(1:k,:),C2(1:k,:),dmin);
disp(['You belongs to class ',num2str(D),' using the LDA classification scheme'])
-------------------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function [A,T]=LoadImage(A,T,Person, N_pictures,Section);
% This function loads images to the matrix A
% input: A - matrix containing images on each column, may be empty
%      T - row containing the class number of each picture, may be empty
%      N_pictures - number of pictures in each person number
%      Section - photo section number
%      Crop- letter indicating the crop section
% Diogo Pereira
% 12/14/2002
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [B,T]=LoadImage(A,T,Person, N_pictures,Section,Crop);
N_class=length(Person);
for i=1:N_class
  for j=1:N_pictures
    im_num1= num2str(Person(i));
    im_num2= num2str(j);
    s=num2str(Section);
    img_name = strcat(im_num1,'-',im_num2,'-',s,'-',Crop);
```

```
    img= imread(img_name,'bmp');
    [dim1,dim2] = size(img);
    x=reshape(img,dim1*dim2,1);
    A=[A x];  % A contains one image in each row
    T=[T Person(i)];% T contains the class of each image
  end
end
clear x;
B=double(A);
Return
```

--------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [C]=meanclass(A,T)
% this function computes the mean of the coordinates of pictures of each class
% input : A - matrix containing the coordinates of each picture
%          in columns
%        T - vector containing the class number of each picture
%        P - vector containing the class numbers that we want to obtain the mean
% output: C - matrix containing the mean of the coordinates of the
%          pictures of each class contained in T
% Diogo Pereira  11/13/02
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```
function [C]=meanclass(A,T,P)
C=[];
for i=1:length(P)
  Temp=P(i);
  TMean=find(T==Temp);
  Mean=mean(A(:,TMean),2);
  C=[C Mean];
end
return
```

--------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function FacemapA computes the square of the difference
% between a subimage and its projection on the facespace;
% the subimage has the size of the eigenface
% input:  W- array (m x n) containing the eigenfaces in columns
%        m dimension of each eigenface, n number of eigenfaces
%        dim1 - dimension 1 of each eigenface, dim2=m/dim1
%        Amean- vector containing the mean of the data used to create the eigenfaces
%        A- array (p x q) containing the image to be analyzed
% output  C- array (r x t) containing the difference between the subimage and its projec-
tion squared
%
% Diogo Pereira
% 10/15/02

73

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [C]=facemapA(W,dim1,Amean,A,T1,T2,T3)
[M,N]=size(A);
[P,L]=size(W);
AF=FFT2(A,256,256); % AF is the FFT of the image
L1=M-dim1+1;
L2=N-(P/dim1)+1;
T5=zeros(L1,L2);
for i=1:L
   U=reshape(W(:,i),dim1,P/dim1);
   UF=AF.*conj(FFT2(U,256,256));
   U=real(IFFT2(UF));
   U=U(1:L1,1:L2);
   U=U-T3(1,i);
   U2=U.*U;
   T5=T5+U2;
end
Amean=reshape(Amean,dim1,P/dim1);
AmeanF=FFT2(Amean,256,256);
T4F=AF.*conj(AmeanF);
T4F=real(IFFT2(T4F));
T4=2*T4F(1:L1,1:L2);
C=T1+T2-T4-T5;
return

--------------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [T1,T2,T3]=facemapini(A,B,W)
% This function computes terms T1, T2 and T3 of the five terms used on the face map
% equation
% input:  A- (M x N)Image where the face is located
%         B- Mean face of the eigenfaces in a column
%         W- (K x L)Matrix containing eigenfaces in columns
%         dim1- vertical dimension of the eigenface
% Diogo Pereira
% last update: 10/15/02
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [T1,T2,T3]=facemapini(A,B,W,dim1);
[M,N]=size(A);
U=dim1;
P=length(B)/dim1;
[K,L]=size(W);
L1=M-dim1+1;
L2=N-P+1;
T1=zeros(L1,L2);
A2=A.*A;
% Computing T1 using FFT2
```

```
A2F=FFT2(A2,256,256);
WindowF=conj(FFT2(ones(dim1,P),256,256));
T1=IFFT2(A2F.*WindowF);
T1=real(T1(1:L1,1:L2));
% Computing T2
B=reshape(B,1,U*P);
T2=B*B';
T2=ones(L1,L2)*T2;
% Computing T3
T3=(B*W);
return

----------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [B]=locatehead(A)
% this function will locate the boundaries of the head
% and will crop the image around the head
% it locates just the horizontal boundaries
% input:   A - original image
% output:  B - cropped image
%
% Diogo Pereira
% last update: 10/11/02
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [B]=locatehead(A);
A=double(A);
[M,N]=size(A);
P=1;
F=N;
Z=zeros(M,1);
D=[A Z]-[Z A];
d=floor(M/6);
P1=find(D(2*d,:)>25);
P2=find(D(3*d,:)>25);
P3=find(D(4*d,:)>25);
% computing P the minimum among P1, P2 and P3
if ~isempty(P1)
   P=P1(1);
end
if ~isempty(P2)
   P=min([P P2(1)]);
end
if ~isempty(P3)
   P=min([P P3(1)]);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
F1=find(D(d*2,:)<-20);
```

```
F2=find(D(d*3,:)<-20);
F3=find(D(d*4,:)<-20);
% computing F the maximum among F1, F2 and F3
if ~isempty(F1)
    F=F1(length(F1));
end
if ~isempty(F2)
    F=max([F F2(length(F2))]);
end
if ~isempty(F3)
    F=max([F F3(length(F3))]);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
B=A(:,P:F);
Return
------------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [D]=classify(PB,P,dmin)
% This function classify PB in classes, each column of P represents a class
% dmin is the minimum distance from each class. If the distance of PB to the
% corresponding class
% is larger than dmin, the anwswer return class O. The distance is the Euclidian distance
% input: PB matrix containig the data to be classified, each column contain the data
%        P matrix containing the coefficient of each class, each column contain the data of
%        one class
%        dmin minimum distance from each class, it is a scalar
% output: D row vector containing the number of the class that each object was classified
% to
%
% Diogo Pereira    07/28/2002
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [D]=classify(PB,P,dmin);
[L,M]=size(PB); %there are M objects
[N,K]=size(P);  %there are K classes
D=zeros(1,M);
for i=1:M
    dif=P-PB(:,i)*ones(1,K);
    [dist,pos]=min(( sum((dif).^2,1) ));
    if dist<dmin
        D(1,i)=pos;
    end
end
return
------------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [C]=distance(A,B)
```

```matlab
% this function computes the distance between the elements in A and the elements in B
%
% input A=  array containing vectors in columns
%      B=  array containing vectors in columns
% output C = array containing the distance between the
% Diogo Pereira
% 11/01/02
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [C]=distance(A,B);
[L,M]=size(A);  %there are M objects
[N,K]=size(B);  %there are K objects
C=zeros(M,K);
for i=1:M
   dif=B-A(:,i)*ones(1,K);
   C(i,:)=sqrt(( sum((dif).^2,1) ));
end
return
```

---------------------------------------------------------------------------------------------

```matlab
function [NV,ND] = sortem(V,D)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%[V,D] = SORTEM(V,D)
%Sorts the columns of V along with the absolute value of the elements of D and
 % eliminates the column of V
% corresponding to zero on the diagonal of D
%
% Diogo Pereira, 2002
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dvec = diag(D);    %obtain the values of the diagonal and insert in a vector
%eliminating the column on V corresponding to zero on D
[I]=find(dvec==0);
dvec(I)=[];
V(:,I)=[];
NV = zeros(size(V));
%sort the elements of dvec in descending order according with the absolute value of dvec
[L,index_dv] = sort(abs(dvec));
index_dv = flipud(index_dv);
dvec=dvec(index_dv);
%insert the elements of dvec on the diagonal
ND=diag(dvec);
%sort the columns of V according with index_dv
NV=V(:,index_dv);
Return
```

---------------------------------------------------------------------------------------------
```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ConverPGMtoBMP -this file loads all the images stored in pgm format and save
% the cropped image in bmp format
```

```
%
% Diogo Pereira
% 12/14/2002
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Person=[1 2 3 4 5 6 8 9 11 12 13 14 15 16]; % Person numbers contained on the files
N_pictures=10; % number of pictures for each person number
Section=1;
load faces W H Amean C %Amg
for i=1:length(Person)
   for j=1:N_pictures
      im_num1= num2str(Person(i)); im_num2= num2str(j);im_num3=num2str(Section);
      img_name = strcat(im_num1,'-',im_num2,'-',im_num3);
      A=readpgm8(img_name); % read pgm files
      %Head=locatehead(A); %locates the head
      B=double(A);
      [m,n]=size(W);
      WI=m/H;
      [T1,T2,T3]=facemapini(B,Amean,W,H);
      Q=facemapA(W,H,Amean,B,T1,T2,T3);
      [M,I]=min(Q);
      [N,yini]=min(M);
      xini=I(yini);
      face=A(xini:(xini+H-1),yini:(yini+WI-1)); % obtain the face on the image
      namefinal=[img_name '-b.bmp'];
      imwrite(face,namefinal,'bmp'); % save the files as bmp
   end
end


%-------------------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% cropface -returns a portion on the image selected
% dragging a windows with a mouse, the size of the window is
% given by H times W
%
% function [B]=cropface(A,W, H)
% input  A = original image, matrix containing the image
%        W = number of pixels wide
%        H = number of pixels high
%output  B = image selected
%
%  Diogo Pereira
% last update: 10/03/2002
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [B]=cropface(A,W,H);
[dim1,dim2]=size(A);
figure('Units','pixels','Position',[100 100 dim2 dim1]);
```

```matlab
imagesc(A);
colormap(gray);
set(gca,'Units','pixels','Position',[0 0 dim2 dim1]);
waitforbuttonpress;
point1 = get(gcf,'CurrentPoint'); % button down detected
rect = [point1(1,1) point1(1,2) W H];
[P] = dragrect(rect);
L=P(1,1); %L is the Left
B=P(1,2); %B is the Bottow
xini=dim1-B-H;
yini=L;
B=A(xini:(xini+H-1),yini:(yini+W-1));
Return

----------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% cropimage - this program crops the images
% The user click on the figure and a window is dragged
% when the user releases the mouse, the portion contained
% within the rectangle is cropped and saved with the same name
% as the original file but with a '-a' added to the file name
% input: file name, typed
% output: image cropped
%
% Diogo Pereira
% last update: 10/03/2002
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
close all
clc
name='';
% data input
while name~='fim'
   name=input('Type the name of the file ->','s');
   temp=name(1,1:3);
   if temp=='fim'
     break
   end
   close all
  a=readpgm8(name); % read pgm files
  W=45;
  H=60;
  b=cropface(a,W,H);
  figure('Units','pixels','Position',[100 100 W H]);
  imagesc(b);
  set(gca,'Units','pixels','Position',[0 0 W H]);
  colormap(gray);
```

```
  namefinal=[name '-a'];
  imwrite(b,namefinal,'bmp');
  name=name(1,1:3);
end
----------------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ProjectPCA
% Diogo Pereira              07/28/02
% This program saves the file faces used in other programs
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
close all
clc
% Testing the routine using the first N_object faces of each class for training and the last
% 10-N_object faces for testing
N_class=16;  %N_Class is the number of classes
N_object=10; %N_object is the number of objects in each class
A=[];
T=[];
% W is an array containing the eigenvectors of the covariance matrix in each column
% Load the training images into A matrix
Person=[1 2 3 4 5 6 8 9 11 12 13 14 15 16];
N_class=length(Person);
N_object=10;
for i=1:N_class
  for j=1:N_object
    im_num1= num2str(Person(i));
    im_num2= num2str(j);
    img_name = strcat(im_num1,'-',im_num2,'-1-a');
     img= imread(img_name,'bmp');
    [dim1,dim2] = size(img);
    x=reshape(img,dim1*dim2,1);
    A=[A x];  % A contains one image in each row
    T=[T Person(i)];
  end
end
clear x;
N_pictures=N_class*N_object; %N_pictures is the total number of pictures on the A ar-
ray
A=double(A);
%show up to 100 images contained in the array A
figure(1)
for i=1:min([N_pictures 100]);
  subplot(10,10,i)
  trial_face=reshape(A(:,i),dim1,dim2);
```

```matlab
    imagesc(trial_face)
    colormap(gray);
    axis off;
end
H=dim1;
% Compute the pca of the data matrix
[W,m,Amean,Ad]=pca(A);
% Compute the matrix P
P=W'*(Ad);
% Computing the centroid of each class=the average of the coefficients
C=[];
for i=1:N_class
    U1=(i-1)*N_object+1;
    U2=U1+N_object-1;
    C(:,i)=mean(P(:,(U1:U2)),2);
end
save faces W Amean H C % stores some data on the file faces
% Loading the testing data
% Load the test images into B matrix
B=[];
T=[];
N_class=14;
Person=[1 2 3 4 5 6 8 9 11 12 13 14 15 16];
N_class=length(Person);
N_object=10;
for i=1:N_class
    for j=1:10
        im_num1= num2str(Person(i));
        im_num2= num2str(j);
        img_name = strcat(im_num1,'-',im_num2,'-5-a');
        img= imread(img_name,'bmp');
        [dim1,dim2] = size(img);
        x=reshape(img,dim1*dim2,1);
        B=[B x];  % B contains one image in each row
        T=[T Person(i)];
    end
end
clear x;
B=double(B);
LB=size(B,2);
B=B-Amean*ones(1,LB);
PB=W'*B;
% Computing the classification
dmin=100000000;
kmax=size(PB,1);
for k=1:kmax
```

```
    [D]=classify(PB(1:k,:),C(1:k,:),dmin);
    error=T-D;
    n_error(k)=sum((error~=0),2);
  end
figure(2)
plot(n_error,'+b')
xlabel('Number of eigenvectors');
ylabel('Number of Errors');
title('Classification errors using 140 test faces');
grid;
end
```

--------------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ProjectLDA
% Diogo Pereira                   10/22/02
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```
clear
close all
clc
% W is an array containing the eigenvectors of the covariance matrix in each column
% Load the training images into A matrix
A=[];
T=[];
Person=[1 2 3 4 5 6 8 9 11 12 13 14 15 16];
N_class=length(Person);
N_object=10;
for i=1:N_class
  for j=1:10
    im_num1= num2str(Person(i));
    im_num2= num2str(j);
    img_name = strcat(im_num1,'-',im_num2,'-1-a');
    img= imread(img_name,'bmp');
    [dim1,dim2] = size(img);
    x=reshape(img,dim1*dim2,1);
    A=[A x];  % A contains one image in each row
    T=[T Person(i)];
  end
end
clear x;
A=double(A);
N_pictures=N_class*N_object; %N_pictures is the total number of pictures on the A array
%show up to 100 image contained in the array A
figure(1)
for i=1:min([N_pictures 100]);
  subplot(10,10,i)
```

```
  trial_face=reshape(A(:,i),dim1,dim2);
  imagesc(trial_face)
  colormap(gray);
  axis off;
end
H=dim1;
% Compute the pca of the data matrix
% Amg=mean(mean(A));
temp=(N_class*N_object) - N_class;
[W,m,Amean,Ad,EVA]=pca(A,temp); % This reduces the dimension to N-c
% Compute the matrix P obtained with the PCA
P=W'*(Ad);
% Compute the Wopt of the Fisher's Linear Discriminant
C1=ones(1,N_object);
C=[];
for temp=1:N_class
   C=[C temp*C1];
%C=[C1 2*C1 3*C1 4*C1 5*C1 6*C1 ...];
end
[Wopt,D]=fld(P,C);
% Compute the projection matrix P obtained with the FLD(Fisher Linear Discriminant)
P=Wopt'*P;
% Computing the centroid of each class=the average of the coefficients
C2=[];
for i=1:N_class
   U1=(i-1)*N_object+1;
   U2=U1+N_object-1;
   C2(:,i)=mean(P(:,(U1:U2)),2);
end
W2=W*Wopt;
save faces2 W2 C2 %Amg
% Load the test images into B matrix
B=[];
T=[];
Person=[1 2 3 4 5 6 8 9 11 12 13 14 15 16];
N_class=length(Person);
N_object=10;
for i=1:N_class
  for j=1:10
     im_num1= num2str(Person(i));
     im_num2= num2str(j);
     img_name = strcat(im_num1,'-',im_num2,'-5-a');
      img= imread(img_name,'bmp');
     [dim1,dim2] = size(img);
     x=reshape(img,dim1*dim2,1);
     B=[B x];  % B contains one image in each row
```

```
      T=[T Person(i)];
   end
end
clear x;
B=double(B);
LB=size(B,2);
B=B-Amean*ones(1,LB);
PB=Wopt'*W'*B;
% Computing the classification
dmin=100000000;
kmax=size(PB,1);
for k=1:kmax
   [D]=classify(PB(1:k,:),C2(1:k,:),dmin);
   error=T-D;
   n_error(k)=sum((error~=0),2);
end
figure
plot(n_error,'-b')
xlabel('Dimension of the transformation');
ylabel('Number of Errors');
title('Classification errors using 140 test faces');
grid;
pause(1)
end;
% % Computing the error in the raw image
% % Load the test images into B matrix
% B=[];
% load faces W H Amean C
%
% for i=1:N_class
%    for j=(N_object + 1):10
%       im_num1= num2str(i);
%       im_num2= num2str(j);
%       img_name = strcat(im_num1,'-',im_num2,'-1');
%        img= readpgm8(img_name);
%        %img=locatehead(img); %locates the head
%        img=double(img);
%        [m,n]=size(W);
%        WI=m/H;
%        [T1,T2,T3]=facemapini(img,Amean,W,H);
%        F=facemapA(W,H,Amean,img,T1,T2,T3);
%        %F=facemap(W,H,Amean,img); % returns F the facemap
%        [M,I]=min(F);
%        [N,yini]=min(M);
%        xini=I(yini);
%        face=img(xini:(xini+H-1),yini:(yini+WI-1)); % obtain the face on the image
```

84

```
%       [dim1,dim2] = size(face);
%       x=reshape(face,dim1*dim2,1);
%       B=[B x];  % B contains one image in each row
%    end
% end
% clear x;
% B=double(B);
% %Normalizing the image
% for i=1:size(B,2)
%     B(:,i)=B(:,i)+Amg-mean(B(:,1));
%     B(:,i)=B(:,i)/max(B(:,i))*256;
% end
%
% temp=N_class*(10-N_object);
% B=B-Amean*ones(1,temp);
% PB=Wopt'*W'*B;
%
% % Computing the classification
% dmin=100000000;
% kmax=size(PB,1);
% T=[];
% for temp=1:N_class
%     for temp2=1:(10-N_object)
%     T=[T temp];
%     end
% end
%
% for k=1:kmax
%    [D]=classify(PB(1:k,:),C2(1:k,:),dmin);
%    error=T-D;
%    n_error(k)=sum((error~=0),2);
% end
% figure
% plot(n_error,'-b')
% xlabel('Dimension of the transformation');
% ylabel('Number of Errors');
% title('Classification errors using 60 test faces');
% grid;
%----------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LocateClassifyA is the advanced version of LocateClassify
% this routine locates a face and classify it according with the PCA method
% but it uses FFT
% input:  keyboard ->name of the file containing the image
%        file faces->W eigenvectors, H high of the image on the database, C centroids of
the classes
```

```
%        Amean mean of the data used to obtain the eigenvectors
%        file faces2-> W2 projection directions, C2 centroids of the classes
% output: number of the class the object was classified to
% Obs.: you need to run the programs ProjectPCA and ProjectLDA before running this
% rotine
% to obtain the files faces and faces2
% respectively
% Diogo Pereira
% last update: 10/09/2002
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all
clear
clc
% obtaining the name of the image
name=input('Type the name of the file ->','s');
T=3;
B=readpgm8(name); % read pgm files
[HI,WI]=size(B);% obtaining the size of the image
% displays the image %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('Units','pixels','Position',[100 100 WI HI]);
imagesc(B);
set(gca,'Units','pixels','Position',[0 0 WI HI]);
colormap(gray)
pause(.8)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp(['Locating the face, wait...'])
% Locates and obtains the face%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
B=locatehead(B); %locates the head
B=double(B);
load faces W H Amean C
load faces2 W2 C2
[m,n]=size(W);
WI=m/H;
[T1,T2,T3]=facemapini(B,Amean,W,H);
Q=facemapA(W,H,Amean,B,T1,T2,T3);
imagesc(Q);
colormap(gray)
[M,I]=min(Q);
[N,yini]=min(M);
xini=I(yini);
face=B(xini:(xini+H-1),yini:(yini+WI-1)); % obtain the face on the image
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% displays the face image%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('Units','pixels','Position',[400 100 WI H]);
imagesc(face)
set(gca,'Units','pixels','Position',[0 0 WI H]);
```

```
colormap(gray)
% classify the face image%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
face=reshape(face,WI*H,1);
face=face-Amean;
%classify using the PCA scheme
PB=W'*face;
dmin=100000000;
kmax=size(PB,1);
k=kmax;
D=classify(PB(1:k,:),C(1:k,:),dmin);
disp(['You belongs to class ',num2str(D),' using the PCA classification scheme'])
%classify using the LDA scheme
PC=(W2)'*face;
dmin=100000000;
kmax=size(PC,1);
k=kmax;
D=classify(PC(1:k,:),C2(1:k,:),dmin);
disp(['You belongs to class ',num2str(D),' using the LDA classification scheme'])
end;
--------------------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% InfraredTotalExitance
% This program computes the Exitance of a blackbody
% plotting it within a wavelength range for a given temperature
%
% Diogo Pereira
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
close all;
clear;
h=6.625e-34;
k=1.38e-23;
c=3e8;
c1=2*pi*h*c^2*10^20
c2=c*h/k*10^6
% c1=3.741e4;
% c2=1.438e4;
% The temperatures to be plotted (insert the desired temperatures on the array T
T=[1150];
LT=length(T);
L1=0.5; % inferior limit for the graph in micrometers
L2=30;  % superior limit for the graph in micrometers
L=logspace(log10(L1),log10(L2),2000);
LL=length(L);
for p=1:LT
    for q=1:LL
```

```
    W(q,p)=c1/( L(q)^5 *(exp( c2/(L(q)*T(p))) - 1));
  end
end
% Generates a Matlab plot
figure
plot(L(:),W(:,1),'b-');
grid;
xlabel('Wavelength [x 10^-6 m]');
ylabel('M - Exitance [Watts/(cm^2 x 10^-6 m)]');
title('Exitance for a blackbody');
% creates the legend
legend('1150K');
% Save the data on a spreadsheet that can be read by excel
filename='ExitanceBlackBody1150K';
M=[L' W(:,1)];
wk1write(filename,M,2,2);
```

-------------------------------------------------------------------------------------------------------

```
function image = readpgm8(filename)
%READPGM8 Read a raw pgm file as a matrix
%
%       IMAGE = READPGM(FILENAME) reads the binary PGM image data from
%       the file named FILENAME and returns the image as a 2-dimensional
%       array of integers IMAGE. Assumes the file is a raw PGM file
%       containing 8-bit unsigned character data to represent pixel values.
%
% Matthew Dailey, 1997
% Modified by Diogo Pereira, 2002
% Open the file
fid = fopen(filename,'r');
% Parse and check the header information.  No # comments allowed.
A = fgets(fid);
if strcmp(A(1:2),'P5') ~= 1
  error('File is not a raw PGM');
end;
A = fgets(fid);
sizes = sscanf(A,'%d');
w = sizes(1);
h = sizes(2);
A = fgets(fid);
max = sscanf(A,'%d');
tlength = w*h;
if max ~= 255
 error('Cannot handle anything but 8-bit graymaps');
end;
% Read the raw data
```

```
[v,count] = fread(fid,inf,'uint8=>uint8'); % this makes v a uint8 instead of double
%if count ~= tlength
% error('File size does not agree with specified dimensions.');
%end;
v=v(1:tlength,1);
% Pack the column vector v into the image matrix
image = reshape(v,w,h)';
fclose(fid);
return
```
-------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CropFaceAuto -This file loads all the images stored in pgm format and save
% the croped image in bmp format
%
% Diogo Pereira
% 12/14/2002
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```
Person=[1 2 3 4 5 6 8 9 11 12 13 14 15 16]; % Person numbers contained on the files
N_pictures=10; % number of pictures for each person number
Section=1;
load faces W H Amean C %Amg
for i=1:length(Person)
    for j=1:N_pictures
        im_num1= num2str(Person(i)); im_num2= num2str(j);im_num3=num2str(Section);
        img_name = strcat(im_num1,'-',im_num2,'-',im_num3);
        A=readpgm8(img_name); % read pgm files
        %Head=locatehead(A); %locates the head
        B=double(A);
        [m,n]=size(W);
        WI=m/H;
        [T1,T2,T3]=facemapini(B,Amean,W,H);
        Q=facemapA(W,H,Amean,B,T1,T2,T3);
        [M,I]=min(Q);
        [N,yini]=min(M);
        xini=I(yini);
        face=A(xini:(xini+H-1),yini:(yini+WI-1)); % obtain the face on the image
        namefinal=[img_name '-b.bmp'];
        imwrite(face,namefinal,'bmp'); % save the files as bmp
    end
end
```
-------------------------------------------------------------------------------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CrossValidate
% This programs test the 2 schemes using cross validation
% without keeping the number of elements per class constant
% Diogo Pereira

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load the images%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=[];  % A contains images in columns
T=[];  % T containt the class number of each picture
Crop='a';
N_pictures=10; % number of pictures for each person number

Person=[1 2 3 4 5 6 8 9 11 12 13 14 15 16]; % Person numbers contained on the files
Section=1;
[A,T]=LoadImage(A,T,Person, N_pictures,Section,Crop);
Section=5;
[A,T]=LoadImage(A,T,Person, N_pictures,Section,Crop);
Section=6;
[A,T]=LoadImage(A,T,Person, N_pictures,Section,Crop);
% Obtain the samples to be used on the testing and remove it from the training images
N_trials=100;
N_samples=100;
APermanent=A;
TPermanent=T;
for i=1:N_trials
    B=[];  % B contains the samples
    TB=[]; % TB contains the class number of each sample
    A=APermanent;
    T=TPermanent;
    N_images=length(T);
    for j=1:N_samples
        Sample=round((N_images-1)*rand)+1;
        B=[B A(:,Sample)];
        A(:,Sample)=[];
        TB=[TB T(Sample)];
        T(Sample)=[];
        N_images=N_images-1;
    end
    % PCA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Compute the pca of the training images A
    [W,m,Amean,Ad]=pca(A);
    % Compute the projection matrix P
    P=W'*(Ad);
    % Compute the centroid of each class contained in Person
    C=meanclass(P,T,Person);
    % Using the testing images B
    Ad=B-Amean*ones(1,size(B,2));
    % Compute the projection matrix P
    P=W'*(Ad);
    % Classify the testing images
    dmin=100000000;
```

```matlab
 [D]=classify(P,C,dmin);
 NZD=find(D~=0);
 D(NZD)=Person(D(NZD));
 error=TB-D;
 n_errorPCA(i)=sum((error~=0),2);
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % LDA %%%%%%%%%%%%%%%%%%%%
 % Compute the pca of the training images
 temp=(N_images-1)*length(Person);
 [W,m,Amean,Ad,EVA]=pca(A,temp); % This reduces the dimension to N-c = 36 -
6=30 the best result was with 29
 % Compute the projection matrix P
 P=W'*(Ad);
 % Compute the projection matrix P obtained with the FLD(Fisher Linear Discriminant)
 [Wopt,D]=fld(P,T);
 P=Wopt'*P;
 % Computing the centroid of each class
 C=meanclass(P,T,Person);
 % Using the testing data
 Ad=B-Amean*ones(1,size(B,2));
 P=W'*(Ad);
 P=Wopt'*P;
 % Classify the testing images
 dmin=100000000;
 [D]=classify(P,C,dmin);
 NZD=find(D~=0);
 D(NZD)=Person(D(NZD));
 error=TB-D;
 n_errorLDA(i)=sum((error~=0),2);
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
figure
L1=min(n_errorPCA);
L2=max(n_errorPCA);
temp1=linspace(0,L2,L2+1);
N1=hist(n_errorPCA,temp1);
plot(temp1,N1,'bd-');
title('Histogram of the number of errors');
xlabel('Number of errors');
ylabel('Number of occurence');
hold on
L1=min(n_errorLDA);
L2=max(n_errorLDA);
temp2=linspace(0,L2,L2+1);
N2=hist(n_errorLDA,temp2);
plot(temp2,N2,'rd-');
```

legend('PCA scheme', 'LDA scheme');

---------------------------------------------------------------------------------------------------

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Function Facemap computes the square of the difference

% between a subimage and its projection on the facespace;

% the subimage has the size of the eigenface

% input:  A- array (m x n) containing the eigenfaces in columns

%        m dimension of each eigenface, n number of eigenfaces

%        dim1 - dimension 1 of each eigenface, dim2=m/dim1

%        Amean- vector containing the mean of the data used to create the eigenfaces

%        B- array (p x q) containing the image to be analyzed

% output  C- array (r x t) containing the difference between the subimage and its

% projection squared

%

% Diogo Pereira

% 09/16/02

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```
function [C]=facemap(A,dim1,Amean,B)
[m,n]=size(A);
dim2=m/dim1;
[p,q]=size(B);
% computing r the number of lines of our facemap
r=p-dim1+1;
% computing t the number of columns of our facemap
t=q-dim2+1;
% creating an empty face map
C=zeros(r,t);
%scanning the image
for L=1:r
   for U=1:t
      % obtaining the subimage PHI
      PHI=B(L:(L+dim1-1),U:(U+dim2-1));   %PHI has dimension dim1 x dim2
      % obtaining the subimage projected on the facespace PHIT
      PHI=reshape(PHI, dim1*dim2,1)-Amean;    %PHI is reshaped to (dim1 x dim2)  x
%1
      Temp=A'*PHI;                % Temp contain the projection of PHI in each eigen-
face
      PHIT=A*Temp;                 % PHIT contain the reconstructed PHI using the ei-
genface
      % obtaining the difference between PHI and PHIT
      C(L,U)=sum((PHI-PHIT).^2);
   end
end
return
```

---------------------------------------------------------------------------------------------------

# LIST OF REFERENCES

[1]     D. Sarkar, "Public Still Unaware of Biometrics," *Federal Computer Week*, Nov 11, 2002, www.fcw.com, last accessed  Nov 16th  2002.

[2]     W. Matthews, "Hands-on Lawmaking," *Federal Computer Week*, Oct 7, 2002, www.fcw.com, last accessed Nov 16th 2002.

[3]     D. Caterinicchia, "DOD Grant Funds Cyberterror Research," *Federal Computer Week*, Oct 10, 2002 www.fcw.com, last accessed  Nov 16th 2002.

[4]     D. Hawkins, "Body of Evidence: Biometrics Turns Your Face, Hand, or Eye into Your Badge of Identity," *U.S. News & World Report*, Washington ; Feb 18, 2002 .

[5]     "Frost & Sullivan: Fallout from September 11 continues – Security concerns catalyse world biometrics markets," *M2 Presswire*, Coventry, Jul 15, 2002.

[6]     L. Mearian, "Visa Eyes Voice Recognition For Online Purchases," *Computerworld*, Framingham, Nov 4, 2002.

[7]     "Nuance Verifier Voice Authentication Software Deployed in Canada Telecommunications Sector," Press Release, www.nuance.com, last accessed Nov 17th 2002.

[8]     M. A. Gips, "Face Recognition Blasted Again," *Security Management*; Arlington; Aug 2002.

[9]     D. Caterinicchia, "Digital MP Adds Facial Recognition," *Federal Computer Week*, Mar 13, 2002, www.fcw.com, last accessed  Nov 16th 2002.

[10]    A. Jain, R. Bolle, S. Pakanti, "Introduction to Biometric" in *Biometrics Personal Identification in Networked Society*, Kluwer Academic Publishers 2002.

[11]    Press release from U.S. Department of Justice, Federal Bureau of Investigation, Aug 10, 1999, www.fbi.gov, last accessed  Nov 18th 2002.

[12]    "IAFIS- Integrated Automated Fingerprint Identification System," www.fbi.gov, last accessed  Nov 18th 2002.

[13]   C. Blank, "At Grocery Checkout, No Wallet Needed," *New York Times*, Jul 25 2002.

[14]   W. Matthews, "Facing Facts: Biometrics, seen as a future cornerstone of security, proves more difficult than feds anticipated," *Federal Computer Week*, Oct 25, 2002, www.fcw.com, last accessed  Nov 16th 2002.

[15]   D. Fuscaldo, "Fingerprint Security is Coming to Personal Computer, Devices," *Wall Street Journal*, Jun 5, 2002.

[16]   F. Prokoski, "History, Current Status, and Future of Infrared Identification,*"* IEEE workshop in *Computer Vision Beyond the Visible Spectrum: Methods and Applications* 2000.

[17]   K.Kondo, N. Kakuta , T. Chinzei, Y. Nasu, T. Suzuki, T. Saito, A. Wagatsuma, H. Ishigaki, K. Mabuchi, "Thermal Rhytmography- Topograms of the Spectral Analysis of Fluctuations in Skin Temperature," *Proceedings of the 23$^{rd}$ Annual EMBS International Conference*, pp. 2812-2815, IEEE 2001.

[18]   Y. V.Gulyaev, A. G. Markov, L. G. Koreneva, P. V. Zakharov, "Dynamical Infra-red Thermography in Humans", pp. 766-771, *IEEE Engineering in Medicine and Biology*, Vol 14,  No 6 , Nov/Dec 1995 .

[19]   M. Burge & W. Burger, "Ear Biometrics in Computer Vision", *Proceedings of the 15th International Conference on Pattern Recognition*, Vol 2, pp. 822-826, 2000.

[20]   "Your Body, Your I.D," *Consumer Reports*, pp. 12-13, Aug 2002.

[21]   D. Caterinicchia, "The Eyes Have It," *Federal Computer Week*, Aug 26, 2002, www.fcw.com, last accessed Nov 16th 2002.

[22]   D. Akin, "Customs Set to Use Iris Scans at Airports", *The Globe and Mail*, Canada's National Newspaper, 2002.

[23]   C. BenAbdelkader, R. Cutler and L. Davis, "Motion-based Recognition of People in EigenGait Space," *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE 2002.

[24]   www.smartnose.com/Presentation.htm, last accessed Nov 19th 2002.

[25]    P. E. Keller, "Overview of Electronic Nose Algorithms," *International Joint Conference on Neural Networks*, Vol 1, pp. 309-312, 1999.

[26]    "Hand Geometry Preferred Biometric in 45 % of All Access Control/Tim and Attendance Applications," Press Release, IR Recognition Systems, Sept 9 2002 www.recogsys.com, last accessed Nov 16th 2002.

[27]    E. Hecht, *Optics*, pp.73-80, Addison Wesley, San Francisco, 2002.

[28]    E.L. Dereniak and G. D. Boreman, *Infrared Detectors and Systems*, John Wiley & Sons, Inc., New York, 1996.

[29]    B. F. Jones, "A Reappraisal of the Use of Infrared Thermal Image Analysis in Medicine", *IEEE Transactions on Medical Imaging*, Vol 17, No 6, pp. 1019 – 1027, Dec 1998.

[30]    M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, Vol 3, No 1, Massachusetts Institute of Technology, 1991.

[31]    R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification*, John Wiley & Sons, 2001.

[32]    C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing*, pp. 54-56, Prentice Hall 1992.

[33]    P.N. Belhumeur, J.P. Hespanha and D.J. Kriegman, "Eigenfaces vs. Fischerfaces: Recognition Using Class Specific Linear Projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 19, No 7, pp. 711 –720, July 1997.

[34]    http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-12.html, last accessed Dec 18th 2002.

THIS PAGE INTENTIONALLY LEFT BLANK

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. John P. Powers
   Chairman, ECE Department
   Naval Postgraduate School
   Monterey, California

4. Centro Tecnico Aeroespacial
   Biblioteca Central
   Sao Jose dos Campos, SP
   Brazil

5. Prof. Monique Fargues
   ECE Department
   Naval Postgraduate School
   Monterey, California

6. Prof. Gamani Karunasiri
   Physics Department
   Naval Postgraduate School
   Monterey, California

7. Prof. Roberto Cristi
   ECE Department
   Naval Postgraduate School
   Monterey, California

8. Diogo C. Pereira
   Centro Tecnico Aeroespacial
   Instituto de Aeronautica e Espaco
   Sao Jose dos Campos, SP
   Brazil